A Summer Internship Report

on

Smart Assistance System for Automated Greetings using Python

Under the supervision of

Prof. (Dr.) Prabhat Kumar

Professor,

Computer Science and Engineering,
National Institute of Technology Patna

At,

Computer Science and Engineering Department,
National Institute of Technology Patna,
Bihar, 800005, India

Submitted by:

Priyanshu Singh

CSE,

MANIT, Bhopal

July 2024

ACKNOWLEDGEMENT

This project is based on automatizing sending greetings on special occasions to contacts of the user using machine python. I would like to thank everyone who helped me to make this project. I would like to thank **Dr. Prabhat Kumar** for his guidance and support throughout this project. It was really nice working under him as I got a chance to enhance my knowledge. I would like to thank **NIT, Patna** for giving me this opportunity. I would also like to thank all the seniors who were always there to support me. I would also like to thank internet for all its knowledge. I would also thank all my friends and family for their constant support. Last but not the least I would like to thank God for his blessings.

THANK YOU

Priyanshu Singh
2311201160
CSE
MANIT, Bhopal

Abstract

In contemporary digital environments, maintaining personal relationships often poses significant challenges. This project aims to develop an advanced Python-based application that automates the process of sending personalized greetings via WhatsApp to users' contacts. The system leverages Python’s extensive libraries for automation, scheduling, and WhatsApp integration, ensuring timely and contextually appropriate greetings for various events such as birthdays, anniversaries, and holidays.

The application seamlessly integrates with users' contact lists and calendars, extracting pertinent information related to birth date and past wishes from users’ chats. By utilizing this database, the program automates the greeting process with diverse and meaningful greetings, thereby adding the personal touch to each communication. The program also features a user-friendly reset and customize options, allowing for user preferences in content of greetings.

The objective of this project is to leverage technology to facilitate the maintenance of personal relationships by reducing the effort required to stay connected, while ensuring that each interaction remains thoughtful and personalized. Through this innovation, we aspire to enhance the quality of digital communication, thereby fostering stronger and more meaningful connections in the increasingly busy lives of our users.

List of Figures

Table of Contents

Introduction

I. Preview

Social media platforms have revolutionized the way and the pace with which individuals connect and expand their network of friends. These digital platforms play a crucial role in fostering relationships and creating opportunities for new connections. In recent times the world has seen, a rise in a no. of social media platforms thus giving the people a no. of options. As a result of this the no. of friends which we are connected with has increased drastically.

Detailed analysis by the team at Kepios shows that there were **5.07 billion** social media users around the world at the start of April 2024, equating to **62.6 percent** of the total global population. Social media user numbers have continued to grow over the past 12 months too, with **259 million** new users joining social media since this time last year. That equates to annualised growth of **5.4 percent**, at an average rate of **8.2** new users **every single second** [1,2,3].

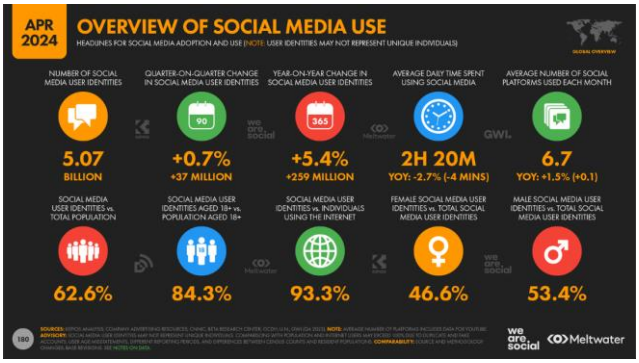
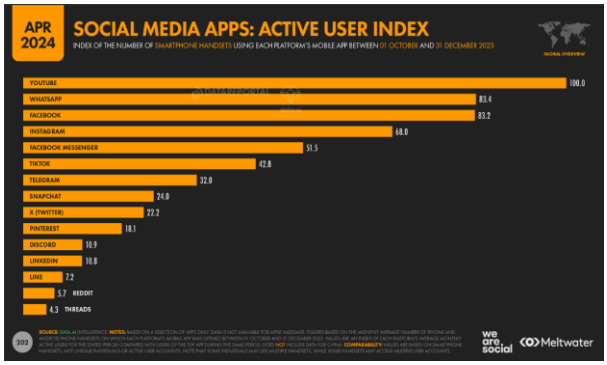
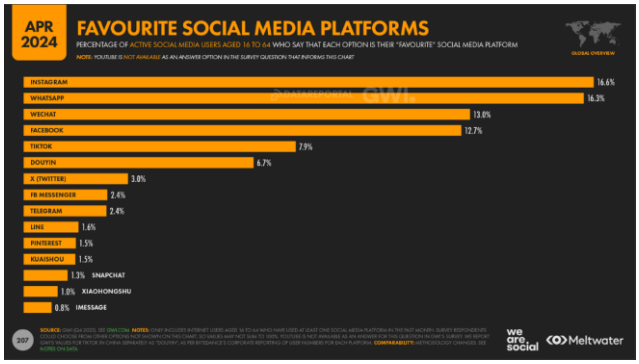


Figure 1. Overview of Social Media Use

Social media is widely used for interacting with people or producing and consuming creative contents. The dominance of sites like Instagram, Facebook, LinkedIn, Snapchat and many more in terms of active users clearly suggest that people like the one to one communication service they provide and also like to explore the various ways in which they provide it.





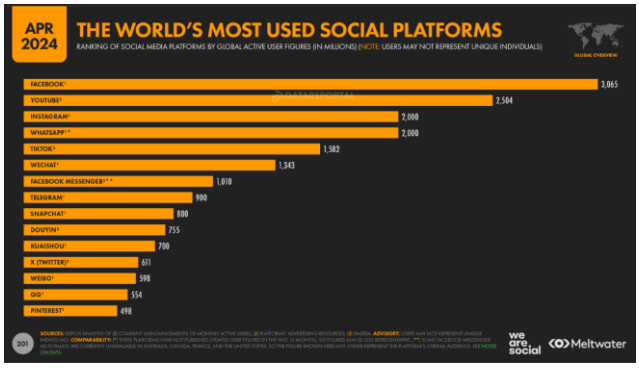


Figure 2. [a] Active User Index
[b] Favourite Social Media Platform
[c] No. of Users of various social media platforms

If we compare the no. of friends in physical world and in cyberspace then according to research commissioned by the **Cystic Fibrosis Trust** an average person has double the number of online

friends than physical ones, which found users of such sites have **121** online friends compared with **55** physical friends [4,5,6].

Therefore, it is quite clear that the no. of friends in our life has increased drastically due to the use of social media.

II. Problem Statement

To develop a system for automating the exchange of greeting messages over WhatsApp.

III. Challenges Involved

Technological advancements have significantly increased the pace of our life. Personal relationships also suffer due to the fast pace of modern life. The lack of quality time spent with family and friends can strain relationships. Additionally, the constant need to stay connected through social media can decrease our attention span and mindfulness, further impacting the quality of our interactions.

We could try to limit our social media interactions by finding automatic ways for less important things like responding to wishes or wishing someone. In our busy life, there is a regular instance where a friend complains us about being unable to wish him on a special occasion. The relationship with online friends is quite fragile and wishing each other on special occasion is a way to keep this friendship on the right track. But, due to a no. of reasons this is not always possible. If the friend resides in a different time zone, then it becomes much harder to wish him on the right time.

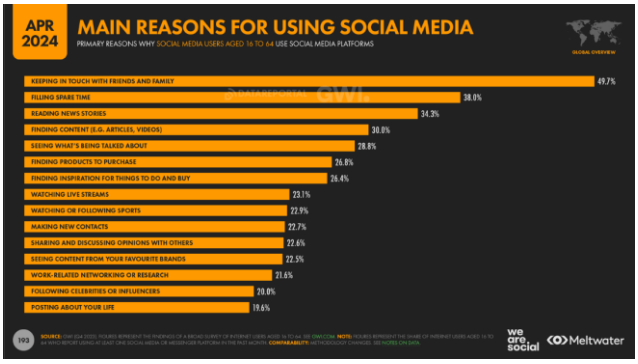


Figure 3. Reason to use social media

We use a no. of social media sites and in most of the cases majority of the people we are connected with are the same in all sites. As a result of this a no. of times we wish the same friend more than once on two different sites for the same thing. This turns out to be an embarrassing situation. Even the person at the receiving end feels bad due to our negligence.

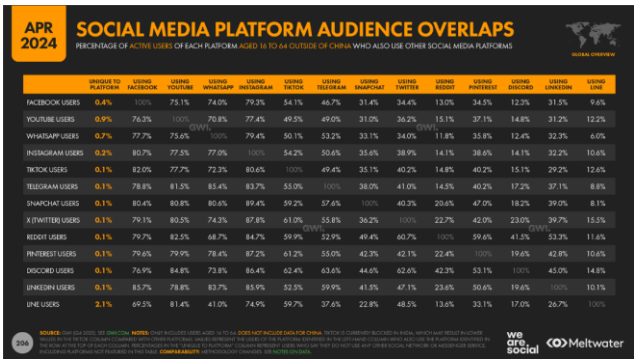
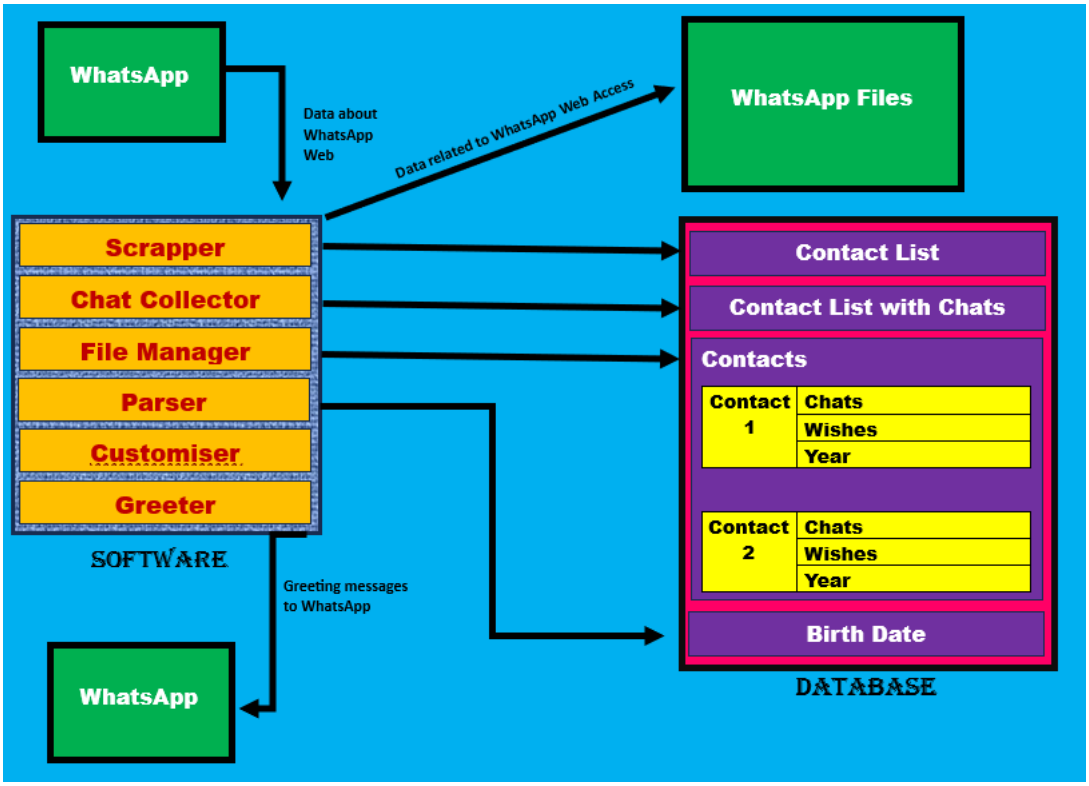


Figure 4. Social Media Platform Audience Overlap

There are also a no. of professional friends who are not personally connected to that much extent, in their case we wish to wish them for an occasion only if they have wished us in the past, but due to our busy life we cannot keep a record of whether they have wished or not. As a result, there are times we do not wish someone in this confusion.

Proposed System

I. Block Diagram



WhatsApp Web is accessed using selenium browser and the related cookies are stored in WhatsApp files in order to avoid any future manual intervention.

The software consists of six parts:

- **Scrapper:** It scraps the name of the contacts added from the WhatsApp web page and stores these contacts inside the database in a text file.
- **Chat Collector:** It collects all the zipped chats which are manually provided by the user and converts it into the unzipped version. It also stores the information about those contacts whose chats have been provided.
- **File Manager:** It creates a folder in the database with the name "Contacts", which consist of a sub folder of the name of all those contacts whose chats are provided. This sub folder contains three text files, the first one containing the chats, and then two empty text files named as "Wishes" and "Year".
- **Parser:** It parses through the chats and obtain the date of birth of that contact and stores this information in the form of a text file "Birth Date". It also stores all the past wishes in the text file wishes for each contact.
- **Customiser:** It provided the facility to customise the wishes anytime we wish.

- **Greeter:** It is used to wish the contact who have birthday today along with informing the user via notification and announcement.

This message is sent back to WhatsApp web without any manual interruption.

The software also provides a no. of reset options which makes using it much easier and more efficient.

II. Control flow:

• Setting up and executing the program:

1. When the programme (**main.py**) is run in the terminal for the first time the permanent function is executed.
2. First of all, "**contact_list.py**" is executed.
   a) The user is asked to scan the QR code.
   b) All the data related to the access of the WhatsApp web get stored in the folder "**WhatsApp Files**".
   c) All the contacts are scrapped and saved in "**Contact List.txt**".
3. The user is then asked to export all the chats to the same folder.
4. Once done "**file_manager.py**" is executed.
   a) It creates a folder of contacts with chat as "**Contact List with Chats.txt**".
   b) It creates a folder "**Contacts**" which contains a subfolder of each contact name.
   c) It unzips the exported chats and store it as text file in respective folders.
5. "**birth_dates.py**" is executed after this.
   a) It stores the birth dates of all the contacts in "**Birth Date**".
   b) It stores all the wishes wished in the past in a file named "**Wishes**" in the respective folders.
6. After this "**customise.py**" gets executed.
   a) It provides options for adding customised wishes based on the category of the contact.
7. "**organise.py**" upon execution creates a dictionary with keys as month and values as another dictionaries having name and date as key-value pair. It is the last step in setting up.
8. "**wish.py**" is executed on the daily basis.
   a) It notifies the user via notification and announcement when it initiates the greeting process.
   b) It uses the cookies in "**WhatsApp Files**" to send the greetings automatically to the contacts.
   c) It creates a file "**Year**" in the respective folder and stores the year of wishing in it.
   d) It also informs if no contact has birthday today or if all the contacts all the contacts having birthday have already been wished.

• Customising wishes:

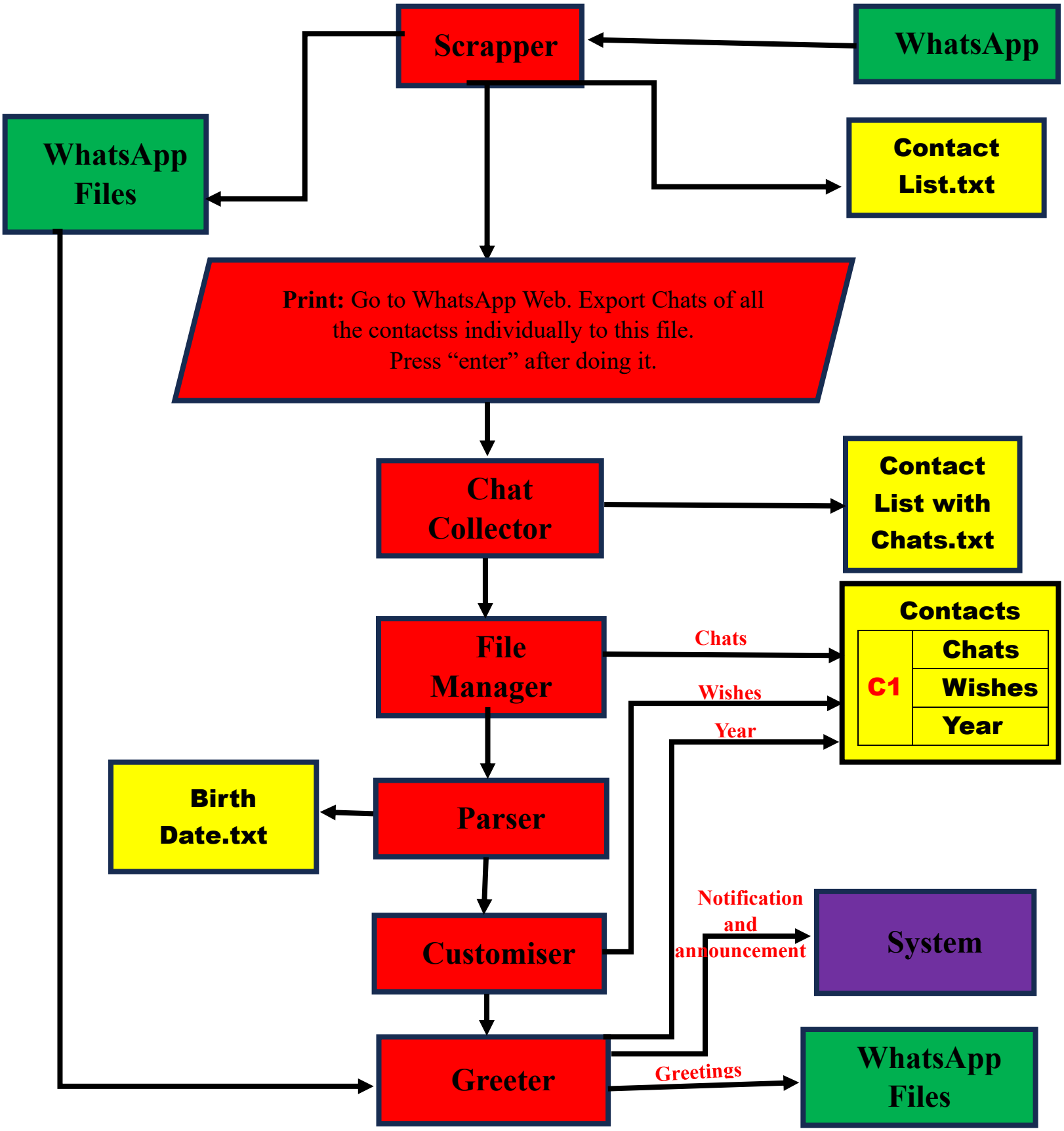
1. When the programme (**customize_individual_wishes.py**) is executed, it provides the user with a list of all the contacts from which the user could choose the contacts.
2. It provides the options for removing a wishing message, replacing a wishing message or adding own message.

• Resetting the wishes:

1. When the programme (**reset.py**) is executed, it provides the user with various options which are:
   - ❖ Complete reset
   - ❖ Partial Reset
   - ❖ Rectifying Birth Dates
   - ❖ Exporting chats of already listed contacts
   - ❖ Manually adding Birthday of already listed contacts
   - ❖ Updating contact list by adding chats
   - ❖ Updating Contact list Manually
   - ❖ Rectifying Contact Names
2. When the user selects any of the above options the function related to that option in "**reset_options.py**" gets automatically executed.
3. Details about reset options:
   a. **Complete reset:** It deletes all the files and the user has to scan the QR code again and do the whole process again.
   b. **Partial Reset:** It deletes all the files and the user has to do the whole process again but the user need not scan the QR code.

c. **Rectifying Birth Dates:** It helps the user in verifying whether the birthdates are correctly entered or not.
d. **Exporting chats of already listed contacts:** It helps in adding the chats of already listed contacts to extract their birthdates.
e. **Manually adding Birthday of already listed contacts:** It helps in manually adding the birthdates of those contacts who are already listed.
f. **Updating contact list by adding chats:** It is used to get all the new contacts and for adding their birthdate by exporting chats.
g. **Updating Contact list Manually:** It is used to add the new contacts manually and for adding their birthdate by exporting chats or manually.
h. **Rectifying Contact Names:** It is used for rectifying Contact n

III. Flowchart:



Implementation

I. User Manual:

```
A. Setting up the program for the first time.

    1. Save this folder such that it is not inside any other folder.

    2. Downloading all the necessary modules:
       Type this in the terminal or powershell: "pip install -r requirements.txt"


    3. Do any one of the following three things to make the program to run when the pc starts:

        a. Task Scheduler:

            # Open Task Scheduler.
            # Click "Create Task".
            # Under the "General" tab, provide a name for the task.
            # Under the "Triggers" tab, click "New" and select "At startup".
            # Under the "Actions" tab, click "New" and choose "Start a program".
            # Browse to your Python executable (e.g., python.exe) and add the path to your script in the "Add arguments" field.
            # Click "OK" to save the task.

        b. Startup Folder:

            # Create a shortcut to your Python script.
            # Press Win + R, type shell:startup, and press Enter to open the Startup folder.
            # Place the shortcut in this folder.

        c.  Run the program "adding_to_startup.py".

    4. Scan the QR code following the instructions on the screen.

    5. Export all the chats to this folder and follow the instructions given in the terminal.

    6. The set up is done, now just restart your pc or run the program "1 main.py", to wish all those who have Birthday today.



B. Customising wishes for certain individuals:

    1. Run the program "customize_individual_wishes.py".

    2. Follow the instructions on the terminal.



C. Resetting the program:

    1. Run the program "reset.py".

    2. Follow the instructions given on the terminal.

    3. Details about reset options:

        a. Complete reset: It deletes all the files and you have to scan the QR code again and do the whole process again.
        b. Partial Reset: It deletes all the files and you have to do the whole process again but you need not scan the QR code.
        c. Rectifying Birth Dates: It helps you in verifying whether the birthdates are correctly entered or not.
        d. Exporting chats of already listed contacts:  It helps in adding the chats of already listed contacts to extract their birthdates.
        e. Manually adding Birthday of already listed contacts: It helps in manually adding the birthdates of those contacts who are already listed.
        f. Updating contact list by adding chats: It is used to get all the new contacts and for adding their birthdate by exporting chats.
        g. Updating Contact list Manually: It is used to add the new contacts manually and for adding their birthdate by exporting chats or manually.
        h. Rectifying Contact Names: It is used for rectifying Contact name.



D. Necessities:

    # Insure that a strong Internet connection is set up for proper functioning.
```

II. Setting up the software and its execution:

- **requirements.txt :**

```
selenium
webdriver-manager
plyer
pywin32
```

- **adding_to_startup.py :**

```python
import os
import win32com.client
import getpass

def create_task(script_path):
    try:
        # Connecting to the Task Scheduler service
        scheduler = win32com.client.Dispatch('Schedule.Service')
        scheduler.Connect()

        # Getting the root task folder
        root_folder = scheduler.GetFolder('\\')

        # Creating a new task definition
        task_def = scheduler.NewTask(0)

        # Creating a trigger that starts the task at startup
        trigger = task_def.Triggers.Create(1)  # 1 = At startup

        # Creating an action that runs the Python script
        action = task_def.Actions.Create(0)  # 0 = Start a program
        action.Path = 'python.exe'
        action.Arguments = f'"{script_path}"'

        # Setting task parameters
        task_def.RegistrationInfo.Description = 'Run Python script at startup'
        task_def.Settings.Enabled = True
        task_def.Settings.StopIfGoingOnBatteries = False
        task_def.Settings.DisallowStartIfOnBatteries = False

        # Registering the task
        task_name = 'PythonStartupScript'
        username = getpass.getuser()
        root_folder.RegisterTaskDefinition(
            task_name,         # Task name
            task_def,          # Task definition
            6,                 # 6 = Create or update
            None,              # User (None = current user)
            None,              # Password (None = current user's password)
            3                  # Logon interactively
        )

    except Exception as e:
        print(f"An error occurred: {e}")


directory_path=os.getcwd()
script_path = os.path.join(directory_path,"1 main.py")
create_task(script_path)

```

- **main.py:**

```python
# 1) Importing the necessary modules
import os
import project_wish

directory_path=os.getcwd()
file_path=os.path.join(directory_path,"Birth Date")


# 2) Running the permanent program for a single time.
if not os.path.exists(file_path):
    project_wish.permanent()


# 3) Running the wishing my contacts program.
project_wish.temporary()
```

- **project_wish.py:**

```python
def permanent():
    # 1) Creating Contact list
    import contact_list

    # 2) Collecting all the zipped chats in the same directory.
    with open("Contacts List.txt","r",encoding="utf-8") as f:
        contact_list=f.read()
        print(contact_list)
    print("End".center(50))
    print("\n")
    print("Go to whatsapp web.")
    print("Export chats of all the contacts individually to this folder.")
    input("Press Enter after doing it.")

    # 3) Unzipping the chats and creating individual folders for each contacts.
    import file_manager

    # 4) Reading the chats and obtaining birth dates and Messages.
    import birth_dates

    # 5) Customising messages for each contact.
    print("Customize ur wishes according to ur wish.".center(50))
    print("Categorize them.".center(50))
    print("\n")

    import customise

def temporary():
    # 1) Arranging the Birth dates according to months.
    # 2) Wishing Happy Birthday.
    import wish
```

- **contact_list.py:**

```python
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.chrome.options import Options
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from webdriver_manager.chrome import ChromeDriverManager
import os

def get_contacts():
    driver = None
    try:
        # Set up Chrome options
        chrome_options = Options()

        current_file_directory = os.getcwd()  # Getting the current directory
        whatsapp_files_directory = os.path.join(current_file_directory, "Whatsapp Files")  # Directory for WhatsApp files

        if not os.path.exists(whatsapp_files_directory):
            os.mkdir(whatsapp_files_directory)

        chrome_options.add_argument(f"user-data-dir={whatsapp_files_directory}")

        # Automatically getting ChromeDriver path and opening WhatsApp Web.
        chrome_driver_path = ChromeDriverManager().install()
        service = Service(executable_path=chrome_driver_path)

        # Initialize WebDriver
        driver = webdriver.Chrome(service=service, options=chrome_options)

        # Open WhatsApp Web
        driver.get("https://web.whatsapp.com")
        print("Opened WhatsApp Web")

        # Wait for the user to scan the QR code and the chat panel to load
        input("Press Enter after scanning the QR code and after WhatsApp Web page is completely loaded...")

        # Wait until the contacts panel is visible
        WebDriverWait(driver, 60).until(
            EC.visibility_of_element_located((By.ID, "pane-side"))
        )

        # Get the contacts panel
        contacts_panel = driver.find_element(By.ID, "pane-side")

        # Get the current visible contacts
        contacts = set()  # Using a set to store contacts to avoid duplicates
        contact_elements = contacts_panel.find_elements(By.CSS_SELECTOR, 'span[dir="auto"]')
        for contact_element in contact_elements:
            # Filter elements based on font and color using JavaScript
            font_color = driver.execute_script("return window.getComputedStyle(arguments[0]).color;", contact_element)

            # Assuming we want to filter for a specific color and font-family (replace with your desired values)
            desired_color1 = "rgb(17, 27, 33)"
            desired_color2="rgb(233, 237, 239)"

            if font_color == (desired_color1):
                name = contact_element.text
                if name:
                    contacts.add(name)

            elif font_color == (desired_color2):
                name = contact_element.text
                if name:
                    contacts.add(name)

        # Storing the contact list into a file
        with open("Contacts List.txt", "w", encoding="utf-8") as f:
            for contact in sorted(contacts):  # Sorting the contacts before writing
                f.write(f"{contact}\n")

    except Exception as e:
        print(f"An error occurred: {e}")
    finally:
        if driver:
            driver.quit()

# Run the function
get_contacts()
```

- **file_manager.py:**

```python
# 1) Importing the necessary Modules.
import zipfile
import os


# 2) Creating a function to unzip file and save it in a different directory.
current_directory_path=os.getcwd()
new_directory_path=os.path.join(current_directory_path,"Contacts")  # Creating a new directory within this directory for Contacts
os.makedirs(new_directory_path, exist_ok=True)

def unzip_file(zip_file_path, extract_to_dir):
    os.makedirs(extract_to_dir, exist_ok=True)

    # Open the zip file in read mode and Extract all the contents into the specified directory
    with zipfile.ZipFile(zip_file_path, 'r') as zip_ref:
        zip_ref.extractall(extract_to_dir)


# 3) Unzipping the file for all the possible contacts.
contact_whose_chats_are_present_=[]
with open("Contacts List.txt","r") as f:  # Obtaining Contact List
    contact=f.read().splitlines()


for e in contact:                           # Unzipping the file
    zip_file_path=os.path.join(current_directory_path,f"WhatsApp Chat with {e}.zip")
    extract_to_dir=os.path.join(new_directory_path,f"{e}")

    if os.path.exists(zip_file_path):
        contact_whose_chats_are_present_.append(e)
        unzip_file(zip_file_path, extract_to_dir)
        os.remove(zip_file_path)            # Removing the Zipped file


with open("Contacts List With Chats.txt","w") as f:  # Obtaining Contact List whose chats are uploaded
        for e in contact_whose_chats_are_present_:
            print(e,file=f)
```

- **birth_dates.py:**

```python
# 1) Importing the necessary modules.
import os



# 2) Obtaining the list of contacts whose files are present.
directory_path=os.getcwd()
contact_list_path=os.path.join(directory_path,"Contacts List With Chats.txt")

with open(contact_list_path,"r") as f:
    contact_list_with_chats=f.read().splitlines()



# 3) Defining the list of elements that should be searched.
wishes=["happy birthday","many many returns of this day","may god bless u with long life"]



# 4) Function for searching for wishes in all the files and getting the birthdates.
def obtain_birthdate(string,contact_name):

    for e in wishes:
        if e in string.lower() and string.split("-")[1].split(":")[0].strip()!=contact_name:
            length=len(string)
            str1=string[0:length-1]
            date=str1.split("-")[0].split(",")[0]
            message=str1.split("-")[1].split(":")[1]
            return(date,message)
        return None



# 5) Searching for wishes in all the files and getting the birthdates.
date_of_birth_file_path=os.path.join(directory_path,"Birth Date")

with open(date_of_birth_file_path,"w",encoding="utf-8") as file1:       # Creating a file with birth dates

    for e in contact_list_with_chats:
        file_path1=os.path.join(directory_path,rf"Contacts\{e}\WhatsApp Chat with {e}.txt")
        file_path_2=os.path.join(directory_path,rf"Contacts\{e}\Wishes.txt")

        with open(file_path_2,"w",encoding="utf-8") as file2:            # Creating a file with wishes.

            with open(file_path1,"r+",encoding="utf-8") as file3:           # Reading the chats
                readlin=1
                flag=False

                while(readlin):
                    readlin=file3.readline()
                    result=obtain_birthdate(readlin,e)

                    if not result == None and flag==False:
                        print(f"{e}:{result[0]}",file=file1)
                        print(f"{result[1].strip()}",file=file2)
                        flag=True

                    elif  not result == None:
                        print(f"{result[1].strip()}",file=file2)
```

- **customise.py:**

```python
# 1) Importing the necessary modules
import os



# 2) Storing the contacts with birthdate in a list.
directory_file_path=os.getcwd()
contact_list_with_birthdate_path=os.path.join(directory_file_path,"Birth Date")

with open(contact_list_with_birthdate_path,"r",encoding="utf-8") as f:
    contact_list_with_birthdate=f.read().splitlines()



# 3) Various options to wish happy Birthday.
w1=["Happy Birthday\n","Happy Birthday Bro\n","Happy Birthday Buddy 👀🥳\n"]
w2=["Happy Birthday\n","Happy Birthday my dear\n","Happy Birthday ❤️\n"]
w3=["Happy Birthday Sir\n","Happy Birthday dear Sir\n","Happy Birthday 🙏\n"]



# 4) Getting the contact list.
contact_list=[]
for e in contact_list_with_birthdate:
    k=e.split(":")[0]
    contact_list.append(k)



# 5) Asking the User to categorise the contacts and customising the Wishes accordingly.
d1={"1":w1,"2":w2,"3":w3}

print("Category 1: Friends and Collegues")
print("Category 2: Family")
print("Category 3: Seniors")
print("Enter 1 for Category 1 and so on.")
print("\n")

for e in contact_list:
    a=input(f"{e}: ")
    file_path=os.path.join(directory_file_path,rf"Contacts\{e}\Wishes.txt")
    with open(file_path,"a",encoding="utf-8") as f:
        f.writelines(d1[a])
```

- **organise.py:**

```python
# 1)# 1) Importing the necessary modules
import os

# 2) Obtaining the necessary path.
directory_file_path=os.getcwd()
contact_list_with_birthdate_path=os.path.join(directory_file_path,"Birth Date")


# 3) Function which organises the data properly.
def organise_data(path):
    def storing_the_data(path):

        with open(path,"r",encoding="utf-8") as f:
            complete_data=f.read()

        data_in_list=complete_data.split("\n")
        data_in_list=list(set(data_in_list))
        data_in_list.remove("")


        return data_in_list


    def arranging_the_data_1(list_of_infos):
        arranged_list=[]

        for e in list_of_infos:
            x=e.split(":")
            y=x[1].split("/")
            y.pop(2)
            y.append(x[0])

            i=0
            while(i<2):
                y[i]=int(y[i])
                i=i+1
            temp=y[0]
            y[0]=y[1]
            y[1]=temp

            arranged_list.append(y)
            arranged_list.sort()

        return(arranged_list)


    def  arranging_the_data_2(data_1):
        data_2={}

        for i in range(1,13):
            data_2[i]=[e for e in data_1 if e[0]==i]


        return (data_2)


    def arranging_the_data_3(data_2):
        data_3={}
        datas1=list(data_2.keys())
        datas2=list(data_2.values())

        for key,value in zip(datas1,datas2) :
            d1={}
```

```
64
65                  for e in value:
66                      key1=e[1]
67                      e.pop(1)
68                      if (key1 in d1.keys()):
69                          d1[key1].append(str(e[1]))
70                      else:
71                          d1[key1]=[(e[1])]
72
73
74                  data_3[key]=d1
75
76              return data_3
77
78
79          #1) Data is stored in a list and all the duplicate data is removed.
80          list_of_infos=list(storing_the_data(path))
81
82          #2) We would store the data in the form of list with each individual elements seperated.
83          #   Sorting the data on the basis of month,date,year and finally name.
84          data_1=arranging_the_data_1(list_of_infos)
85
86          #3) We would form a dictionary based on month
87          #   The dictionary is made in such a way that it stores the information of each month
88          data_2=arranging_the_data_2(data_1)
89
90          #4) The dictionary is formed with each month as key and a list as value which contains key value pair of name and date of birth
91          data_3=arranging_the_data_3(data_2)
92          return data_3
93
94
95
96      # 4) Calling the function to organiise the data.
97      organised_data=organise_data(contact_list_with_birthdate_path)
98
99
100
101     # 5) Creating a function to be called from another module.
102     def get_data():
103         return organised_data
```

- **wish.py:**

```python
# 1) Obtaining the necessary data from previous module.
import organise
data=organise.get_data()

# 2) Obtaining the necessary path.
import os
directory_file_path=os.getcwd()
contact_path=os.path.join(directory_file_path,"Contacts")



# 3) Notification function
def notify(contact_name):
    from plyer import notification
    import time

    notification.notify(
        title='Birthday Wish',
        message=f'Wishing Happy Birthday to {contact_name}',
        app_name='Wishify',
        timeout=5
    )
    time.sleep(2)


# 4) Announcement function
def announce(contact_name):
    import win32com.client
    import time
    speaker = win32com.client.Dispatch("SAPI.SpVoice")
    s = f'Wishing Happy Birthday to {contact_name}'
    speaker.Speak(s)
    time.sleep(2)


# 5) Messaging function
def message():
    from selenium import webdriver
    from selenium.webdriver.common.by import By
    from selenium.webdriver.chrome.service import Service
    from selenium.webdriver.chrome.options import Options
    from selenium.webdriver.common.keys import Keys
    from selenium.webdriver.support.ui import WebDriverWait
    from selenium.webdriver.support import expected_conditions as EC
    from webdriver_manager.chrome import ChromeDriverManager
    import os
    import time
    import random

    def write_message(contact_name):
```

```python
    try:
        # 1) Set up Chrome options
        chrome_options = Options()

        current_file_directory = os.getcwd()  # Getting the address of current directory
        whatsapp_files_directory = os.path.join(current_file_directory, "Whatsapp Files")  # Creating a directory for WhatsApp files

        if not os.path.exists(whatsapp_files_directory):
            os.mkdir(whatsapp_files_directory)

        chrome_options.add_argument(f"user-data-dir={whatsapp_files_directory}")


        # 2) Automatically getting ChromeDriver path and opening WhatsApp Web.
        chrome_driver_path = ChromeDriverManager().install()
        service = Service(executable_path=chrome_driver_path)

        # Initialize WebDriver
        driver = webdriver.Chrome(service=service, options=chrome_options)

        # Open WhatsApp Web
        driver.get("https://web.whatsapp.com")
        print("Opened WhatsApp Web")

        # Wait for the whatsapp web to be downloaded.
        time.sleep(20)


        # 3) Getting the message.
        file_path=os.path.join(current_file_directory,rf"Contacts\{contact_name}\Wishes.txt")
        with open(file_path,"r",encoding="utf-8") as f:
            mess=f.read().splitlines()
        message=random.choice(mess)



        # 4)Searching for the contact
        search_box = WebDriverWait(driver, 30).until(
            EC.presence_of_element_located((By.XPATH, '//div[@contenteditable="true"][@data-tab="3"]'))
        )
        search_box.clear()
        search_box.send_keys(contact_name)
        search_box.send_keys(Keys.RETURN)



        # 5) Wait for the chat to load
        chat_loaded = WebDriverWait(driver, 30).until(
            EC.presence_of_element_located((By.XPATH, f'//span[@title="{contact_name}"]'))
        )
```

```
                chat_loaded.click()



                # 6)Send message
                message_box = WebDriverWait(driver, 30).until(
                    EC.presence_of_element_located((By.XPATH, '//div[@contenteditable="true"][@data-tab="10"]'))
                )
                message_box.send_keys(message)
                message_box.send_keys(Keys.RETURN)
                time.sleep(5)
                driver.quit()



            except Exception as e:
                print(f"An error occurred: {e}")


        # Run the function
        write_message(contact_name)


    # 6) Get date/month/year
    def date():
        import time
        t1=time.localtime()
        t2=time.strftime("%d:%m:%Y",t1)
        l=t2.split(":")
        return l


    # 7) Wishing everyone and storing the date.
    l=date()
    date=int(l[0])
    month=int(int(l[1]))
    year=int(int(l[2]))

    list1=data[month]
    flag1=False
    flag2=False

    if list1 != []:

        if date in list1.keys():
            for e in list1[date]:
                try:
                    flag1=True
                    contact_name=e
                    file_path=os.path.join(contact_path,rf"{contact_name}\Year.txt")
```

```
                    with open(file_path,"a+",encoding="utf-8") as f:
                        f.seek(0)
                        a=f.read().splitlines()

                        if str(year) not in a:
                            notify(contact_name)
                            announce(contact_name)
                            message()
                            flag2=True
                            f.seek(2)
                            f.write(f"{year}\n")
                except Exception as b:
                    print("An error occured.",b)


    if flag1==False:
        print("No one has Birthday today.".center(50))
    elif flag2==False:
        print("All those who have Birthday today have already been wished.".center(50))
```

III. Customising the Wishes:

- **customize.individual_wishes.py:**

```python
import os

directory_path=os.getcwd()
file_path1=os.path.join(directory_path,"Birth Date")
file_path2=os.path.join(directory_path,"Contacts")

l1=[]



def custom_wish(l):
    for idx,e in enumerate(l,start=1):
        print(f"{idx}: {e}")
    print()

    l_modified=[]
    print("Enter 0 if u want to remove the message.")
    print("Enter 1 if u want to replace the message.")
    print("Enter 2 if u want to keep the message as it is.")

    for e in l:
        print()
        print(e)
        print()
        k1=int(input("Enter ur response: "))
        if k1==1:
            q1=input("Enter the wish: ")
            l_modified.append(q1)
        elif k1==2:
            l_modified.append(e)

    print()
    print("Write additional wishes which u would like to add.")
    print("Once done enter 0.")
    print()

    while(True):
        q2=input("Enter the wish: ")
        print()
        if q2=="0":
            break
        l_modified.append(q2)


    return l_modified



with open(file_path1,"r",encoding="utf-8") as f:
    data=f.read().splitlines()

for e in data:
    a=e.split(":")[0]
    l1.append(a)

for idx,e in enumerate(l1,start=1):
    print(f"{idx}: {e}")



print()
print("Enter the index of the contact whose wishes have to be customised.")
print("Enter 0 when done.")
print()


while(True):
    x=int(input("Enter the index: "))
    if x==0:
        break
    file_path3=os.path.join(file_path2,rf"{l1[x-1]}\Wishes.txt")

    with open(file_path3,"r",encoding="utf-8") as f:
        data1=f.read().splitlines()

        data2=custom_wish(data1)

    with open(file_path3,"w",encoding="utf-8") as f:
        for e in data2:
            print(e,file=f)
    print()
```

IV. Resetting the Program:

- **reset.py:**

```python
# 1) Importing the necessary file.
import reset_options as ro

# 2) Creating a function for importinf the correct function.
def func(n):
    if(n==1):
        ro.complete_reset()
    elif(n==2):
        ro.partial_reset()
    elif(n==3):
        ro.birth_date_checking()
    elif(n==4):
        ro.chatless_conatcts()
    elif(n==5):
        ro.chatless_contacts_manually()
    elif(n==6):
        ro.update_contact()
    elif(n==7):
        ro.update_contact_manually()
    elif(n==8):
        ro.correcting_name()
    elif(n!=0):
        print("Invalid Input")


# 3) Giving choices to the user.

print("Reset".center(100,"."))
print("\n")

list1=["Complete reset","Partial Reset","Rectifying Birth Dates","Exporting chats of already listed contacts", "Manually adding Birthday of already listed contacts","Updating contact list by adding chats","Updating Contact list Manually","Rectifying Contact Names"]
for idx,e in enumerate(list1,1):
    print(f"{idx}: {e}")

print()
print("Choose the option which u wish to use.")
print("Give the input as 0 if u wish to exit.")
print()

a=1
while(a):
    a=int(input("Enter your choice: "))
    print()
    func(a)
    print()


print("Reset completed.".center(100,"."))
```

- **reset_options.py:**

```
# 0) Importing the important files
import os
import shutil
import time
import zipfile



# 1) COMPLETE RESET

def complete_reset():
    try:
        if(os.path.exists("Whatsapp Files")):
            shutil.rmtree("Whatsapp Files")

        if(os.path.exists("Contacts")):
            shutil.rmtree("Contacts")

        if(os.path.exists("Contacts List.txt")):
            os.remove("Contacts List.txt")


        if(os.path.exists("Contacts List With Chats.txt")):
            os.remove("Contacts List With Chats.txt")

        if(os.path.exists("Birth Date")):
            os.remove("Birth Date")


    except Exception as e:
        print("There was some error in resetting.")



# 2) PARTIAL RESET

def partial_reset():
    try:
        if(os.path.exists("Contacts")):
            shutil.rmtree("Contacts")

        if(os.path.exists("Contacts List.txt")):
            os.remove("Contacts List.txt")


        if(os.path.exists("Contacts List With Chats.txt")):
            os.remove("Contacts List With Chats.txt")

        if(os.path.exists("Contacts List Without Chats.txt")):
            os.remove("Contacts List Without Chats.txt")

        if(os.path.exists("Birth Date")):
            os.remove("Birth Date")


    except Exception as e:
        print("There was some error in resetting.")
```

```
    except Exception as e:
        print("There was some error in resetting.")



# 3) Recheck the Birth date of your contacts.

def birth_date_checking():
    directory_path=os.getcwd()
    file_path=os.path.join(directory_path,"Birth Date")

    with open(file_path,"r",encoding="utf-8") as f:
        data=f.read()
        data_processed=data.splitlines()

    for idx,e in enumerate(data_processed,start=1):
        print(f"{idx} {e}")

    print()
    print("Enter 0 if u have confirmed.")
    print()

    a=int(input("Enter your response: "))
    if a==0:
        print("The Birthdates have been confirmed.")
    else:
        while(a):
            print("Enter date and month in two digits.")
            date=input("Enter the date: ")
            month=input("Enter the month: ")

            contact_path=os.path.join(directory_path,"Contacts")
            year_path=os.path.join(contact_path,f"{data_processed[a-1].split(':')[0]}Year.txt")

            if os.path.exists(year_path):
                with open(year_path,"r",encoding="utf-8") as f:
                    f.seek(0)
                    a1=f.read().splitlines()
                    year_1=time.strftime("%Y")
                    if(year_1 in a1):
                        a1.remove(year_1)

                with open(year_path,"w",encoding="utf-8") as f:
                    for e in a1:
                        print(e,file=f)

            year=data_processed[a-1].split("/")[2]

            value=data_processed[a-1].split(":")[0] + f":{date}/{month}/{year}"
            data_processed[a-1]=value
            a=int(input("Enter your response: "))


        with open(file_path,"w",encoding="utf-8") as f:
            for e in data_processed:
                print(e,file=f)
```

```
# 4) Getting the list of contacts whose chats are not exported.

def chatless_conatcts():
    directory_path=os.getcwd()
    file_path1=os.path.join(directory_path,"Contacts List.txt")
    file_path2=os.path.join(directory_path,"Contacts List With Chats.txt")
    file_path3=os.path.join(directory_path,"Contacts")

    with open(file_path1,"r",encoding="utf-8") as f1:
        data1=set(f1.read().splitlines())


    with open(file_path2,"r",encoding="utf-8") as f2:
        data2=set(f2.read().splitlines())

    data3=list(data1.difference(data2))
    for idx,e in enumerate(data3,start=1):
        print(f"{idx} {e}")

    if data3 !=[]:
        print()
        print("Export the chats of the following to this folder in the form of zipped file.")
        print("Once u are done enter 0.")
        print()

        while(True):
            a=int(input("Enter 0 when done: "))
            if(a==0):
                break
            else:
                print("Invalid Input.")

        os.rename("Contacts List.txt","Contacts List1.txt")
        os.rename("Contacts List With Chats.txt","Contacts List With Chats1.txt")
        os.rename("Birth Date","Birth Date1")

        with open("Contacts List.txt","w",encoding="utf-8") as f:
            for e in data3:
                print(e,file=f)

        # Unzipping the chats and creating individual folders for each contacts.
        import file_manager

        # Reading the chats and obtaining birth dates and Messages.
        import birth_dates

        # Customising messages for each contact.
        print()
        print("Customize ur wishes according to ur wish.".center(50))
        print("Categorize them.".center(50))
        print("\n")

        import customise

        os.remove("Contacts List.txt")

        with open("Birth Date","r",encoding="utf-8") as f:
            a1=f.read()
        with open("Birth Date1","a",encoding="utf-8") as f1:
            f1.write(a1)
        os.remove("Birth Date")
        os.rename("Birth Date1","Birth Date")
```

```
    with open("Contacts List With Chats.txt","r",encoding="utf-8") as f:
        a1=f.read()
    with open("Contacts List With Chats1.txt","a",encoding="utf-8") as f1:
        f1.write(a1)
    os.remove("Contacts List With Chats.txt")
    os.rename("Contacts List With Chats1.txt","Contacts List With Chats.txt")


    os.rename("Contacts List1.txt","Contacts List.txt")

else:
    print("Chats of all the contacts which are presented in the contact list have been already exported.")




# 5) Getting the list of contacts whose chats are not exported manually.

def chatless_contacts_manually():
    directory_path=os.getcwd()
    file_path1=os.path.join(directory_path,"Contacts List.txt")
    file_path2=os.path.join(directory_path,"Contacts List With Chats.txt")
    file_path3=os.path.join(directory_path,"Contacts")

    with open(file_path1,"r",encoding="utf-8") as f1:
        data1=set(f1.read().splitlines())


    with open(file_path2,"r",encoding="utf-8") as f2:
        data2=set(f2.read().splitlines())

    data3=list(data1.difference(data2))
    for idx,e in enumerate(data3,start=1):
        print(f"{idx} {e}")

    if data3 !=[]:
        print()
        print("Enter the sequence no. of the contact which u would like to manually add.")
        print("Once u are done enter 0.")
        print()

        a=1

        additional_contacts=[]
        additional_contacts_with_birth_date=[]
        while(a):
            a=int(input("Enter the sequence no: "))
            if a==0:
                break
            date=input("Enter the date in two digits: ")
            month=input("Enter the month in two digits: ")
            print()

            year=time.strftime("%Y")
            data_to_be_added=f"{data3[a-1]}:{date}/{month}/{year[2:]}"

            additional_contacts.append(data3[a-1])
            additional_contacts_with_birth_date.append(data_to_be_added)

        with open("Contacts List Without Chats.txt","a+",encoding="utf-8") as f:
```

```
        with open("Contacts List Without Chats.txt","a+",encoding="utf-8") as f:
            f.seek(2)
            for e in additional_contacts:
                print(e,file=f)
                file_path4=os.path.join(file_path3,e)
                file_path5=os.path.join(file_path4,"Wishes.txt")
                os.makedirs(file_path4,exist_ok=True)

                # Various options to wish happy Birthday.
                w1=["Happy Birthday\n","Happy Birthday Bro\n","Happy Birthday Buddy 🙌🎁\n"]
                w2=["Happy Birthday\n","Happy Birthday my dear\n","Happy Birthday ❤️\n"]
                w3=["Happy Birthday Sir\n","Happy Birthday dear Sir\n","Happy Birthday 🙏\n"]

                # Asking the User to categorise the contacts and customising the Wishes accordingly.
                d1={"1":w1,"2":w2,"3":w3}


                print("Category 1: Friends and Collegues")
                print("Category 2: Family")
                print("Category 3: Seniors")
                print("Enter 1 for Category 1 and so on.")
                a=input(f"{e}: ")
                print("\n")

                with open(file_path5,"a",encoding="utf-8") as f2:
                    f2.writelines(d1[a])

        with open("Birth Date","a+",encoding="utf-8") as f:
            f.seek(2)
            for e in additional_contacts_with_birth_date:
                print(e,file=f)

    else:
        print("Chats of all the contacts which are presented in the contact list have been already exported")




# 6) Adding New contacts and their date of birth

def update_contact():
    directory_path=os.getcwd()
    file_path1=os.path.join(directory_path,"Contacts List.txt")
    file_path2=os.path.join(directory_path,"Contacts List With Chats.txt")
    file_path3=os.path.join(directory_path,"Contacts")

    os.rename("Contacts List.txt","Contacts List1.txt")
    os.rename("Contacts List With Chats.txt","Contacts List With Chats1.txt")
    os.rename("Birth Date","Birth Date1")

    import contact_list

    with open(file_path1,"r",encoding="utf-8") as f1:
        data1=set(f1.read().splitlines())


    with open("Contacts List1.txt","r",encoding="utf-8") as f2:
        data2=set(f2.read().splitlines())

    data3=list(data1.difference(data2))
```

```
    with open(file_path1,"w",encoding="utf-8") as f1:
        for e in data3:
            print(e,file=f1)



    # Collecting all the zipped chats in the same directory.
    with open("Contacts List.txt","r") as f:
        contact_list=f.read()
        print(contact_list)
    print("End".center(50))
    print("\n")
    print("Go to whatsapp web.")
    print("Export chats of all the contacts individually to this folder.")
    input("Press Enter after doing it.")

    # Unzipping the chats and creating individual folders for each contacts.
    import file_manager

    # Reading the chats and obtaining birth dates and Messages.
    import birth_dates

    # Customising messages for each contact.
    print("Customize ur wishes according to ur wish.".center(50))
    print("Categorize them.".center(50))
    print("\n")

    import customise

    with open("Birth Date","r",encoding="utf-8") as f:
        a1=f.read()
    with open("Birth Date1","a",encoding="utf-8") as f1:
        f1.write(a1)
    os.remove("Birth Date")
    os.rename("Birth Date1","Birth Date")


    with open("Contacts List With Chats.txt","r",encoding="utf-8") as f:
        a1=f.read()
    with open("Contacts List With Chats1.txt","a",encoding="utf-8") as f1:
        f1.write(a1)
    os.remove("Contacts List With Chats.txt")
    os.rename("Contacts List With Chats1.txt","Contacts List With Chats.txt")


    with open("Contacts List.txt","r",encoding="utf-8") as f:
        a1=f.read()
    with open("Contacts List1.txt","a",encoding="utf-8") as f1:
        f1.write(a1)
    os.remove("Contacts List.txt")
    os.rename("Contacts List1.txt","Contacts List.txt")



# 7) Adding New contacts and their date of birth both manually.

def update_contact_manually():
    print("1 Adding name and chat of the contact.")
    print("2 Adding name and birthdate of the contact.")
    print()
    x=int(input("Enter your choice: "))
```

Smart Assistance System for Automated Greetings using Python

```
    directory_path=os.getcwd()
    file_path1=os.path.join(directory_path,"Contacts List.txt")
    file_path2=os.path.join(directory_path,"Contacts List With Chats.txt")
    file_path3=os.path.join(directory_path,"Contacts")



    if x==1:

        print("Enter the name of the contact u wish to add.")
        print("Once u are done,enter 0.")
        print()

        l1=[]
        while(True):
            a=input("Enter the Name of the contact: ")
            if a=="0":
                break

            l1.append(a)

        os.rename("Contacts List.txt","Contacts List1.txt")
        os.rename("Contacts List With Chats.txt","Contacts List With Chats1.txt")
        os.rename("Birth Date","Birth Date1")

        with open("Contacts List.txt","w",encoding="utf-8") as f:
            for e in l1:
                print(e,file=f)

        with open("Contacts List.txt","r",encoding="utf-8") as f:
            contact_list=f.read()
            print(contact_list)
        print("You".center(50))
        print("\n")
        print("Go to whatsapp web.")
        print("Export chats of all the contacts individually to this folder.")
        input("Press Enter after doing it.")

        # Unzipping the chats and creating individual folders for each contacts.
        import file_manager

        # Reading the chats and obtaining birth dates and Messages.
        import birth_dates

        # Customising messages for each contact.
        print("Customize ur wishes according to ur wish.".center(50))
        print("Categorize them.".center(50))
        print("\n")

        import customise


        with open("Birth Date","r",encoding="utf-8") as f:
            a1=f.read()
        with open("Birth Date1","a",encoding="utf-8") as f1:
            f1.write(a1)
        os.remove("Birth Date")
        os.rename("Birth Date1","Birth Date")


        with open("Contacts List With Chats.txt","r",encoding="utf-8") as f:
            a1=f.read()
        with open("Contacts List With Chats1.txt","a",encoding="utf-8") as f1:
```

```
        with open("Contacts List With Chats1.txt","a",encoding="utf-8") as f1:
            f1.write(a1)
        os.remove("Contacts List With Chats.txt")
        os.rename("Contacts List With Chats1.txt","Contacts List With Chats.txt")


        with open("Contacts List.txt","r",encoding="utf-8") as f:
            a1=f.read()
        with open("Contacts List1.txt","a",encoding="utf-8") as f1:
            f1.write(a1)
        os.remove("Contacts List.txt")
        os.rename("Contacts List1.txt","Contacts List.txt")



    elif x==2:

        print("Enter the name of the contact u wish to add.")
        print("Once u are done,entrer 0.")
        print()

        l1=[]
        while(True):
            a=input("Enter the Name of the contact: ")
            if a=="0":
                break

            l1.append(a)

        with open("Contacts List.txt","a",encoding="utf-8") as f:
            for e in l1:
                print(e,file=f)


        additional_contacts_with_birth_date=[]
        for e in l1:
            print(f"Input the Birthdate of {e}")

            date=input("Enter the date in two digits: ")
            month=input("Enter the month in two digits: ")
            print()

            year=time.strftime("%Y")
            data_to_be_added=f"{e}:{date}/{month}/{year[2:]}"

            additional_contacts_with_birth_date.append(data_to_be_added)

        with open("Contacts List Without Chats.txt","a+",encoding="utf-8") as f:
            f.seek(2)
            for e in l1:
                print(e,file=f)
                file_path4=os.path.join(file_path3,e)
                file_path5=os.path.join(file_path4,"Wishes.txt")
                os.makedirs(file_path4,exist_ok=True)

                # Various options to wish happy Birthday.
                w1=["Happy Birthday\n","Happy Birthday Bro\n","Happy Birthday Buddy 🙌🤩\n"]
                w2=["Happy Birthday\n","Happy Birthday my dear\n","Happy Birthday 💛\n"]
                w3=["Happy Birthday Sir\n","Happy Birthday dear Sir\n","Happy Birthday 🙏\n"]

                # Asking the User to categorise the contacts and customising the Wishes according
                d1={"1":w1,"2":w2,"3":w3}
```

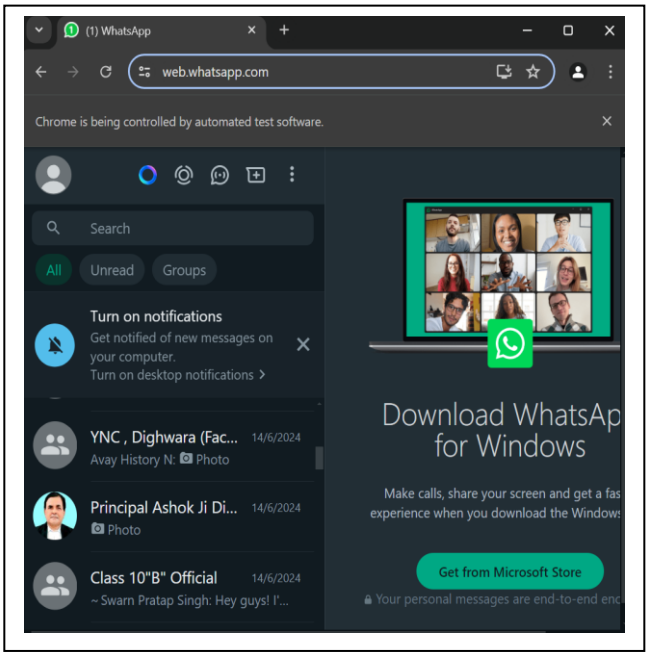
```
                print("Category 1: Friends and Collegues")
                print("Category 2: Family")
                print("Category 3: Seniors")
                print("Enter 1 for Category 1 and so on.")
                a=input(f"{e}: ")
                print("\n")

                with open(file_path5,"a",encoding="utf-8") as f2:
                    f2.writelines(d1[a])

        with open("Birth Date","a+",encoding="utf-8") as f:
            f.seek(2)
            for e in additional_contacts_with_birth_date:
                print(e,file=f)



    else:

        print("Invalid input.")
        exit()



# 8) Rectifying Contact Names
def correcting_name():
    directory_path=os.getcwd()
    file_path1=os.path.join(directory_path,"Contacts List.txt")
    file_path2=os.path.join(directory_path,"Contacts List With Chats.txt")
    file_path3=os.path.join(directory_path,"Contacts List Without Chats.txt")
    file_path4=os.path.join(directory_path,"Birth Date")
    file_path5=os.path.join(directory_path,"Contacts")

    with open("Birth Date","r",encoding="utf-8") as f:
        data_raw=f.read().splitlines()
        data=[]

        for e in data_raw:
            k=e.split(":")[0]
            data.append(k)

    for idx,e in enumerate(data,start=1):
        print(f"{idx}: {e}")

    print()
    print("Choose the contact name u wish to change.")
    print("Enter 0 when u are done.")
    print()

    d1={}
    while(True):
        a=int(input("Enter your  response: "))

        if a==0:
            break
```

```
        old_name=data[a-1]
        print("Old Name of the Contact:",old_name)
        new_name=input("Enter the correct name: ")
        print("New Name of the Contact:",new_name)
        d1[old_name]=new_name
        file_path=os.path.join(file_path5,old_name)
        file_path_new=os.path.join(file_path5,new_name)
        if os.path.exists(file_path):
            file_path_x=os.path.join(file_path,f"WhatsApp Chat with {old_name}.txt")
            if os.path.exists(file_path_x):
                os.remove(file_path_x)

            os.rename(file_path,file_path_new)

    with open(file_path4,"r",encoding="utf-8") as f:
        data1=f.read()

    for key in d1.keys():
        dat1=data1.replace(key,d1[key])

    with open(file_path4,"w",encoding="utf-8") as f:
        f.write(dat1)



    with open(file_path2,"r",encoding="utf-8") as f:
        data1=f.read()

    for key in d1.keys():
        dat1=data1.replace(key,d1[key])

    with open(file_path2,"w",encoding="utf-8") as f:
        f.write(dat1)



    with open(file_path1,"r",encoding="utf-8") as f:
        data1=f.read()

    for key in d1.keys():
        dat1=data1.replace(key,d1[key])

    with open(file_path1,"w",encoding="utf-8") as f:
        f.write(dat1)

    if os.path.exists(file_path3):
        with open(file_path3,"r",encoding="utf-8") as f:
            data1=f.read()

        for key in d1.keys():
            dat1=data1.replace(key,d1[key])

        with open(file_path3,"w",encoding="utf-8") as f:
            f.write(dat1)
```

Result and Analysis

I. Setting up the file for the first time:

```
DevTools listening on ws://127.0.0.1:49952/devtools/browser/3416c58b-bb3f-48e2-9309-f456e2c4d131
Opened WhatsApp Web
Press Enter after scanning the QR code and after WhatsApp Web page is completely loaded...Created TensorFlow Lite XNNPACK delegate for CPU.
```




```
DevTools listening on ws://127.0.0.1:49952/devtools/browser/3416c58b-bb3f-48e2-9309-f456e2c4d131
Opened WhatsApp Web
Press Enter after scanning the QR code and after WhatsApp Web page is completely loaded...Created TensorFlow Lite XNNPACK delegate for CPU.

+91 77658 70536
Boss family group
Class 10"B" Official
Hp Whatsapp
Mummy Jio
Papa BSNL
Prashasti Singh
Principal Ashok Ji Dighwara
Priyanshu Jio
Sherkhan
YNC , Dighwara (Faculty)
ZOOLOGY (20-23)
Zoology ( 22-25)
Zoology (2021-24)

                End


Go to whatsapp web.
Export chats of all the contacts individually to this folder.
Press Enter after doing it.
```

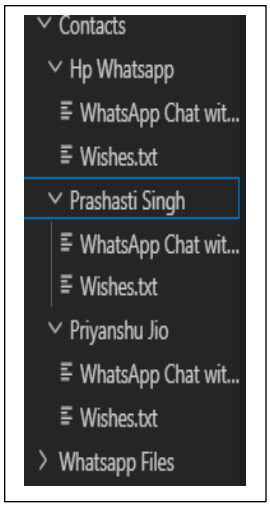
Contacts List.txt

```
1  +91 77658 70536
2  Boss family group
3  Class 10"B" Official
4  Hp Whatsapp
5  Mummy Jio
6  Papa BSNL
7  Prashasti Singh
8  Principal Ashok Ji Dighwara
9  Priyanshu Jio
10 Sherkhan
11 YNC , Dighwara (Faculty)
12 ZOOLOGY (20-23)
13 Zoology ( 22-25)
14 Zoology (2021-24)
```

Contacts List With Chats.txt

```
1  Hp Whatsapp
2  Prashasti Singh
3  Priyanshu Jio
```

```
DevTools listening on ws://127.0.0.1:49952/devtools/browser/3416c58b-bb3f-48e2-9309-f456e2c4d131
Opened WhatsApp Web
Press Enter after scanning the QR code and after WhatsApp Web page is completely loaded...Created TensorFlow Lite XNNPACK delegate for CPU.

+91 77658 70536
Boss family group
Class 10"B" Official
Hp Whatsapp
Mummy Jio
Papa BSNL
Prashasti Singh
Principal Ashok Ji Dighwara
Priyanshu Jio
Sherkhan
YNC , Dighwara (Faculty)
ZOOLOGY (20-23)
Zoology ( 22-25)
Zoology (2021-24)

                End


Go to whatsapp web.
Export chats of all the contacts individually to this folder.
Press Enter after doing it.
    Customize ur wishes according to ur wish.
                Categorize them.


Category 1: Friends and Collegues
Category 2: Family
Category 3: Seniors
Enter 1 for Category 1 and so on.


Prashasti Singh: 1
Priyanshu Jio: 2
          No one has Birthday today.
```



Birth_date.txt

```
1  Priyanshu Jio:28/06/24
2  Beti Singh:21/06/24
3
```

II. Wishing the contacts:

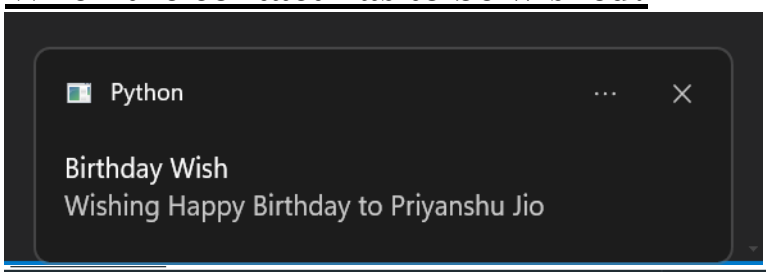
- **When no one has birthday on that date:**

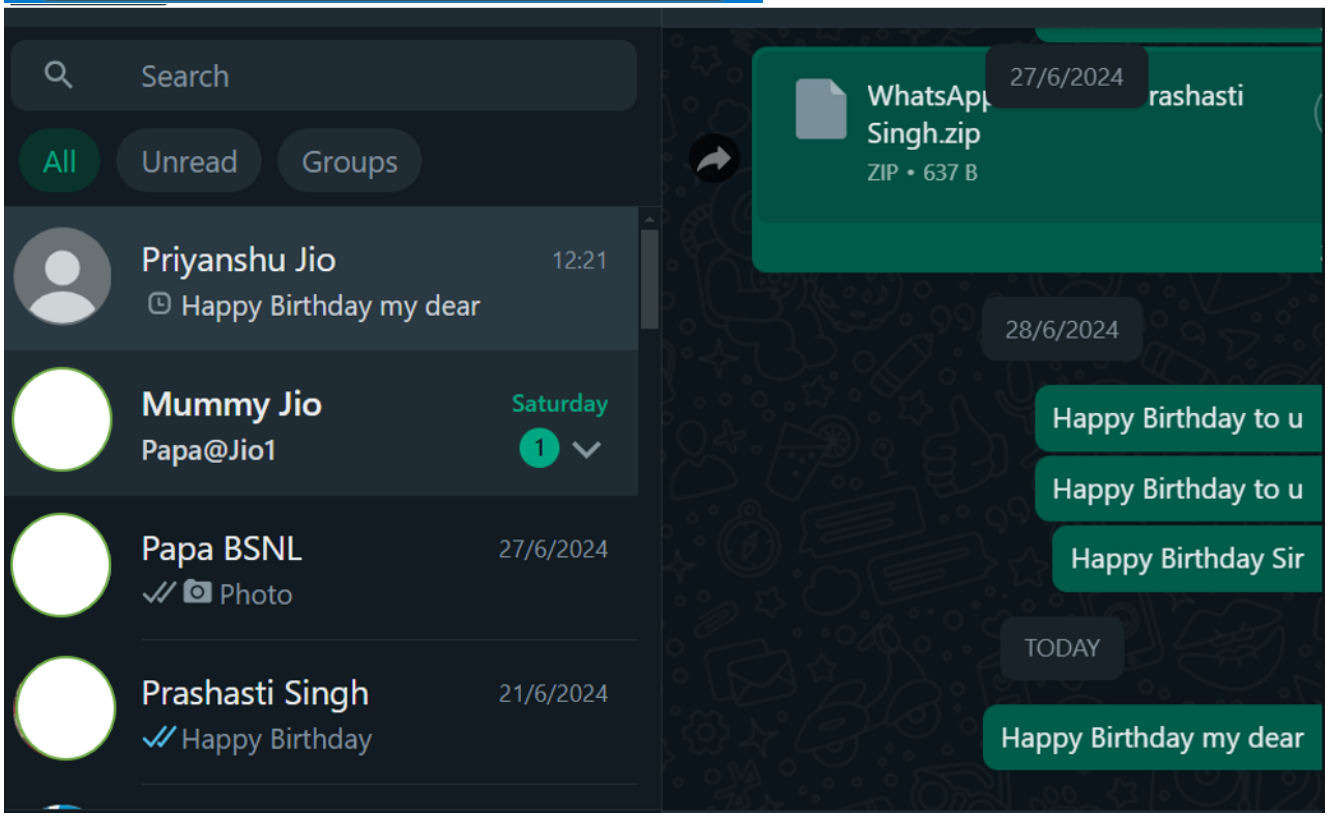
```
No one has Birthday today.
```

- **When the contact has been already greeted:**

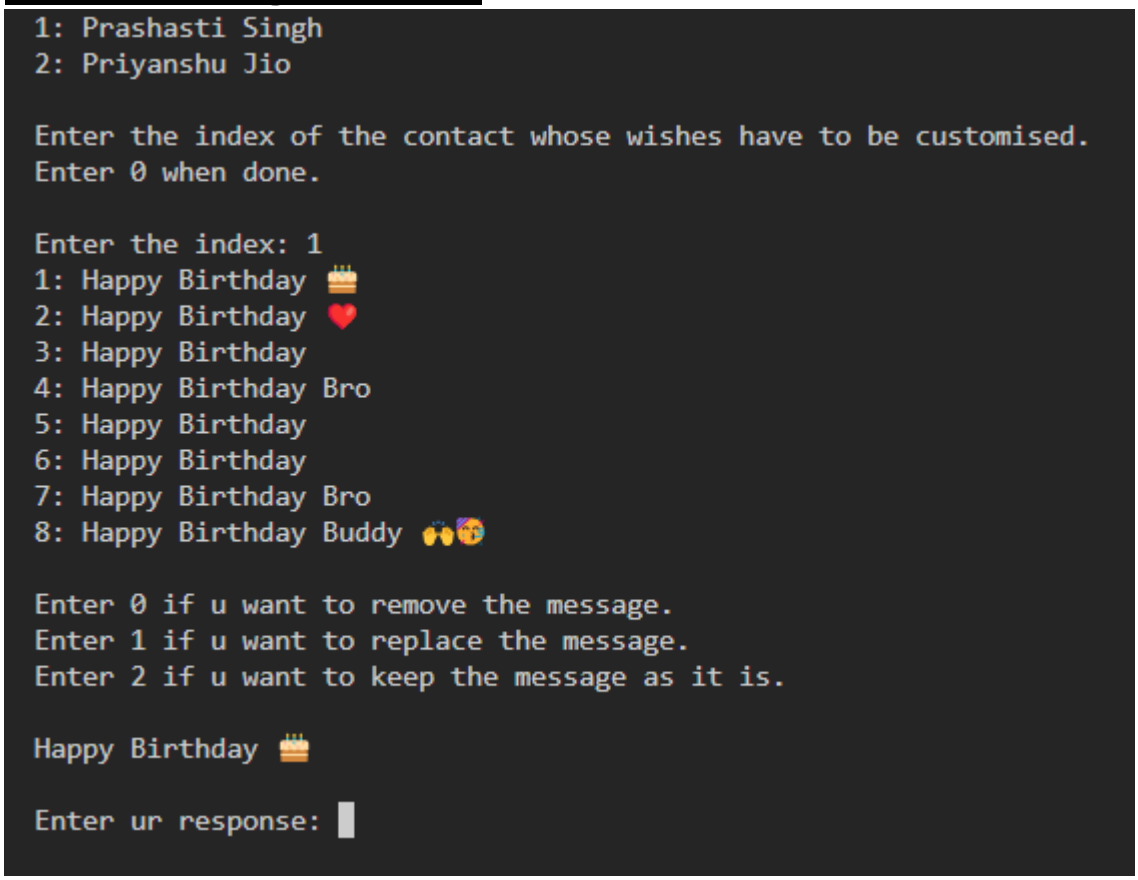
```
All those who have Birthday today have already been wished.
```

- **When the contact has to be wished:**





III. Customising Wishes:



IV. Resetting the program:

Conclusion

We can use this solution for wishing our friends birthday by using WhatsApp, after setting it up according to the instructions given in user manual file.
Once done we can either run the program daily or we can program it to automatically run once the computer turns on. We can also customise the wishes rectify the contact names and birth dates.
There are a few obstructions in proper running of the program:

a) Accessing WhatsApp web through selenium does not allows to scrap all the contacts directly due to privacy issues.
   We can get 14 to 15 contacts but we need to add the rest of the contacts manually.
   We can also use WhatsApp API to access all the contacts but it requires permission from WhatsApp and a business account.
b) The program is written for windows, so a few changes are required to run it in mac.
c) The program is able to scrap the contacts only when the colour is not customised by the user and only the default dark and light themes are used.
d) The code for running the program automatically after startup works only when the system has default settings. In case the program shows error, manual ways are written in the user manual which should be followed.

References

[1] Global Social Media Statistics – DataReportal-Global Digital Insights
[2] datareportral.com
[3] https://datareportal.com/social-media-users#:~:text=Social%20media%20user%20numbers%20have,new%20users%20every%20single%20second
[4] The Guardian
[5] theguardian.com
[6] https://www.theguardian.com/media/2011/may/09/social-network-users-friends-online