MID-TERM REPORT

Social Media Utility Tool

Submitted by

SHRIYANSHI BARANWAL (181500691)
PRASHANT ASTHANA (181500487)
TANISHQ TRIPATHI (181500751)

In partial fulfillment of the

Requirements

For the degree of

Bachelors of Technology

In

Computer Science

Supervised By

Dr. Manoj Varshney

Department of Training and Development

GLA University, Mathura

CONTENT

DECLARATION

We hereby declare that the work which is being presented in the MINI PROJECT “**Social Media Utility Tool**”, will be fulfilled of the requirements for Mini Project viva voce, is an authentic record of my work carried under the supervision of **Dr. Manoj Varshney**.

Course: Bachelor of Technology

Branch: Computer Science

Year: III

Semester: 6th

Name of Candidate (Roll No):

- SHRIYANSHI BARANWAL (181500691)
- PRASHANT ASTHANA (181500487)
- TANISHQ TRIPATHI (181500751)

Signature of Candidates:

ACKNOWLEDGMENT

We express our sincere indebtedness towards our mentor, **Dr. Manoj Varshney**, Computer Science & Engineering. GLA University, Mathura for his invaluable guidance, suggestions, and supervision throughout the work. Without his kind patronage and guidance, the project would not have taken shape. We would also like to express our gratitude and sincere regards for his kind approval of the project, time-to-time counseling, and pieces of advice.

We would also like to thank our HOD **Mr. Anand Singh Jalal** Department of Computer Science & Engineering. GLA University, Mathura for her expert advice and counseling from time to time.

We owe sincere thanks to all the faculty members in the department of Computer Science & Engineering for their kind guidance and encouragement from time to time.

Shriyanshi Baranwal(181500691)

Prashant Asthana(181500487)

Tanishq Tripathi(181500751)

ABSTRACT

In our project we are creating an application to combine all basic applications in a single utility. With this one can easily schedule a WhatsApp message, send email to a particular person directly, search anything on Google or can directly download a YouTube video also. The general problems to be addressed by the Social Media Utility Tool is the following:

a. To provide reliable and easy to use Social Media tools that manage an individual user's need.

While a number of commercially available and public domain products have solved these problems, the Social Media Utility Tool provides a solution with certain functional improvements over existing tools. Specific problem areas are the following:

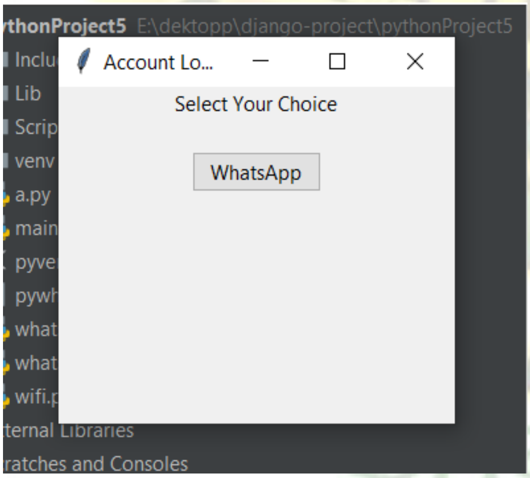
a. Scheduling message sending on a certain time.
b. Sending email direct through the application.
c. Search anything from google.
d. Search any video on YouTube.

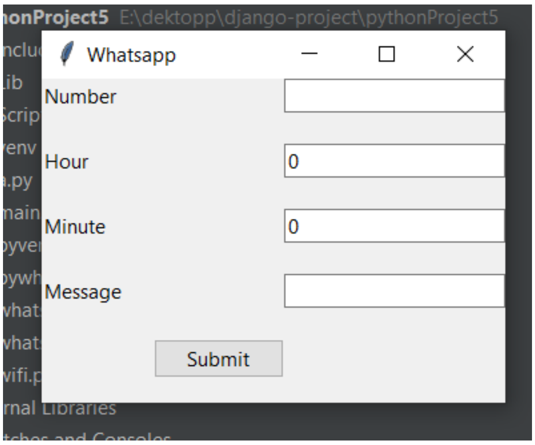
Here we will be using python programming language for frontend and Tkinter frame work for GUI.

INTRODUCTION

For our project we are creating an application to combine all basic applications in a single utility. Through the application you can easily schedule a WhatsApp message, send email to a particular person, directly search anything on Google or can directly download a you tube video also.

1. **Home Page-**Here we will add 2 options, login and register.
2. **Login Page/Register Page-** By using our credentials i.e. Id & valid password we can login to the welcome page. Here we can also create the new password in case we forgot the old one and can also redirect to the previous page i.e. home page.
3. **Welcome Page-**Here we will and the options for WhatsApp, Gmail, google search, YouTube search.





Number

Hour 0

Minute 0

Message

LANGUAGE USED

PYTHON:

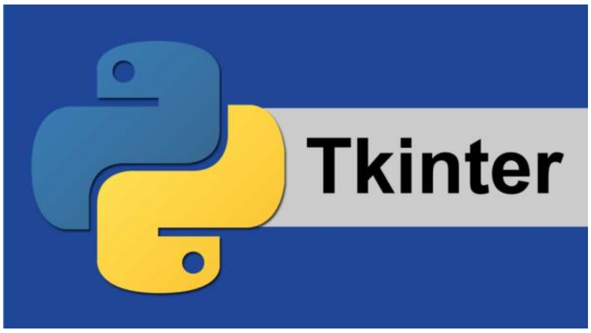
Python is an interpreted, high-level and general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library. Python is meant to be an easily readable language. Its formatting is visually uncluttered, and it often uses English keywords where other languages use punctuation. Unlike many other languages, it does not use curly brackets to delimit blocks, and semicolons after statements are optional. It has fewer syntactic exceptions and special cases than C or Pascal.

Python is a general-purpose programming language. Hence, we can use the programming language for developing both desktop and web applications. Also, we can use Python for developing complex scientific and numeric applications. Python is designed with features to facilitate data analysis and visualization.

Reason for selecting Python:

- Readable and Maintainable Code
- Multiple Programming Paradigms
- Compatible with Major Platforms and Systems
- Robust Standard Library
- Many Open Source Frameworks and Tools
- Simplify Complex Software Development
- Adopt Test Driven Development

ABOUT TKINTER (LIBRARY IN PYTHON)



Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

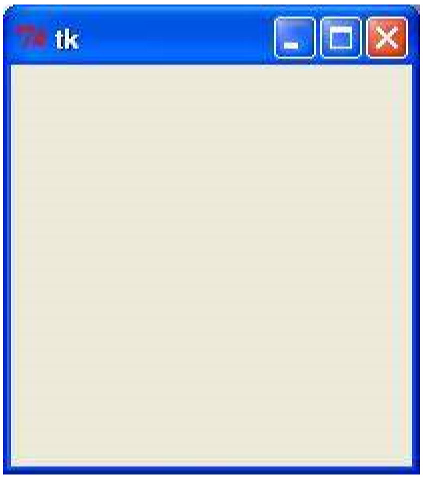
Creating a GUI application using Tkinter is an easy task. All you need to do is perform the following steps −

- Import the *Tkinter* module.
- Create the GUI application main window.
- Add one or more of the above-mentioned widgets to the GUI application.
- Enter the main event loop to take action against each event triggered by the user.

Example:

```
#!/usr/bin/python
import Tkinter
top = Tkinter.Tk()
# Code to add widgets will go here...
top.mainloop()
```

This would create a following window −



OBJCETIVE &REQUIREMENTS

Objective: -

The main objective of creating such an application is to combine all basic applications in a single utility. Through the application one can easily schedule a WhatsApp message, send email to a particle person, directly search anything on Google or can directly download a YouTube video also. The Social Media Utility Tool addresses the requirements of a particular customer. One needs a tool with a collection of features not available in any single existing tool. This aspect of the problem is one of combining a set of existing features in a way that the features have yet to be combined in another product.

Hardware: -

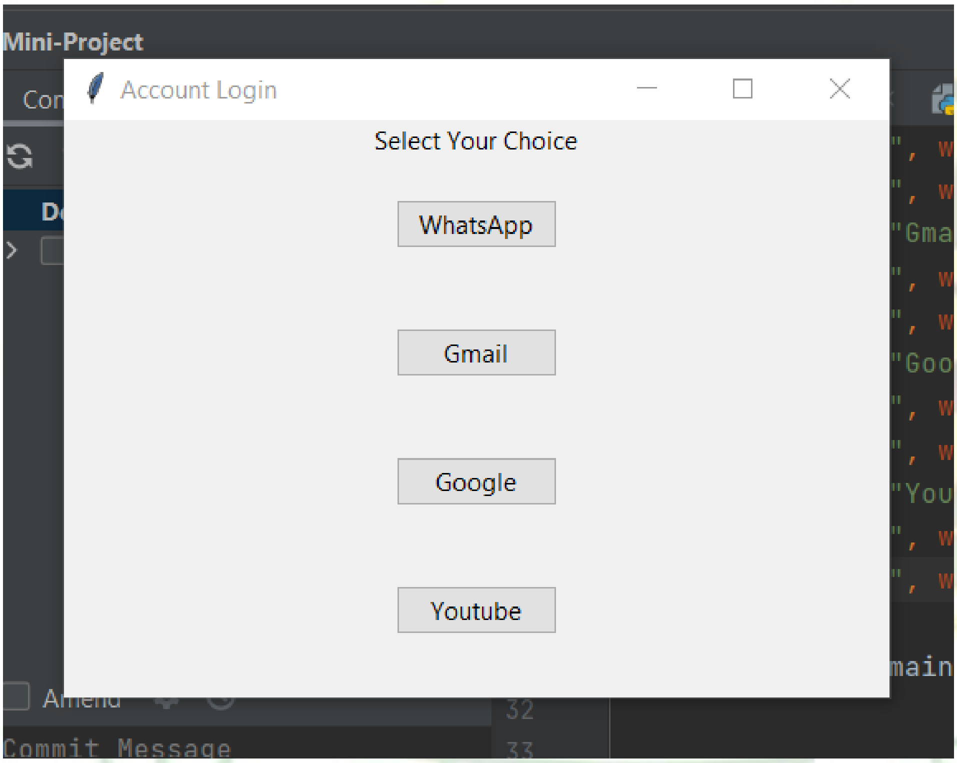
OS: - Mac OSX 10.10+ / Windows 7/8/10
RAM: - 4GiB Recommended
Hard Drive: -1GiB of available disk Space minimum (700MiB for IDE)
Screen Resolution: -1280x800 screen resolution
Processor: -At least i3 8th gen 1.6GHz or Faster Processor

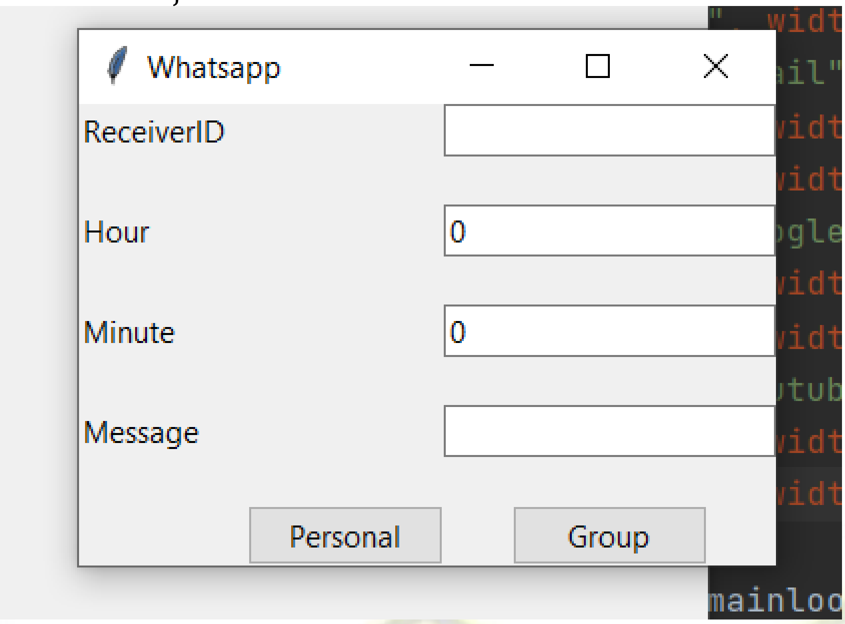
Software: -

PyCharm CE
Visual Studio Code

PROGRESS & SCRRENSHOTS

We have successfully created WhatsApp screen, google screen and YouTube screen of the application. Till now we can send personal message or group message through WhatsApp web and search anything on google and any video on YouTube.



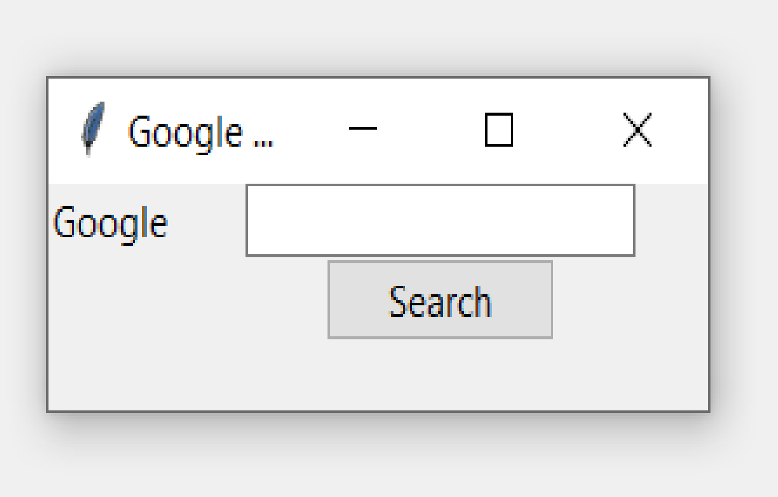


ReceiverID

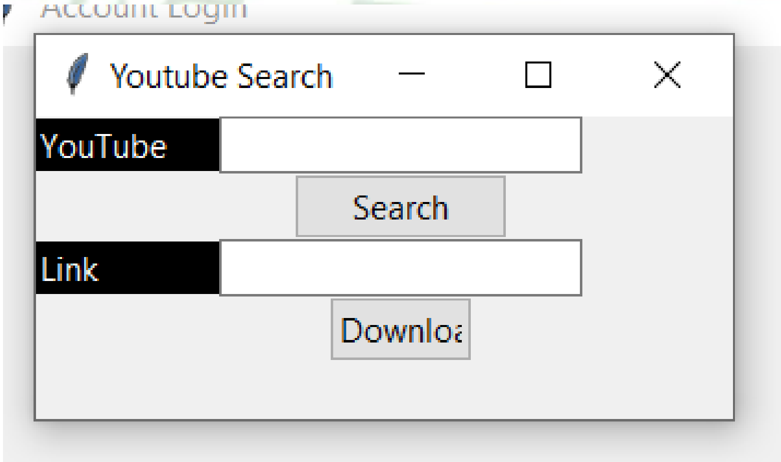
Hour 0

Minute 0

Message



Google

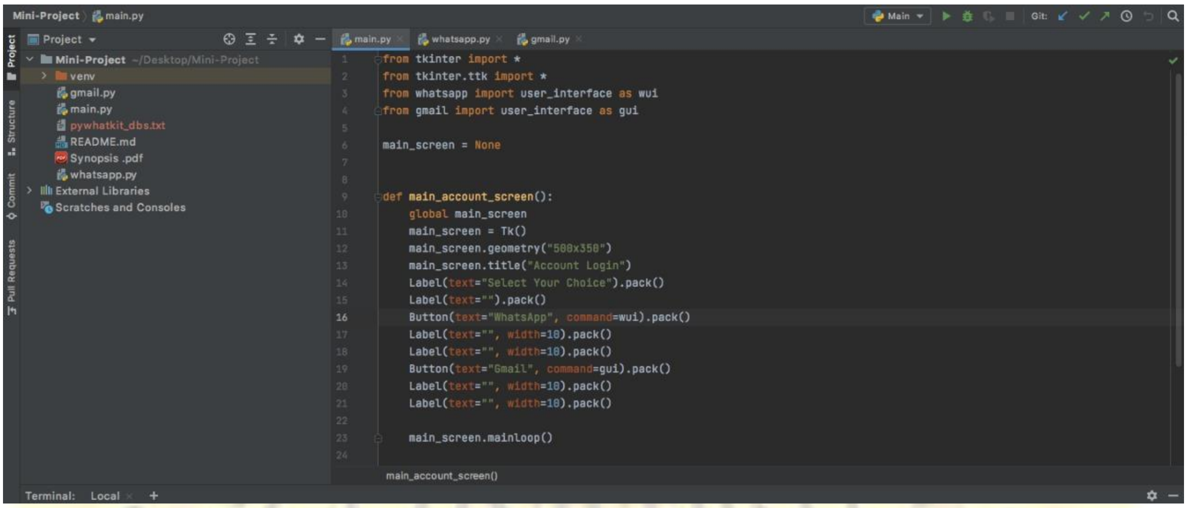


YouTube

Link

CODE SCRRENSHOTS

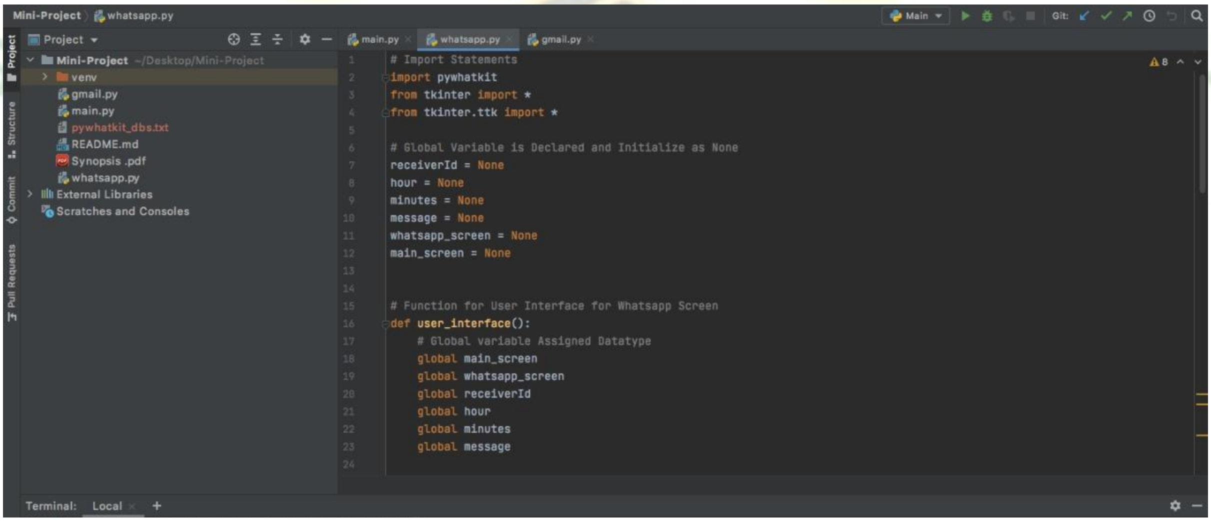
Main.py



```python
from tkinter import *
from tkinter.ttk import *
from whatsapp import user_interface as wui
from gmail import user_interface as gui

main_screen = None


def main_account_screen():
    global main_screen
    main_screen = Tk()
    main_screen.geometry("500x350")
    main_screen.title("Account Login")
    Label(text="Select Your Choice").pack()
    Label(text="").pack()
    Button(text="WhatsApp", command=wui).pack()
    Label(text="", width=10).pack()
    Label(text="", width=10).pack()
    Button(text="Gmail", command=gui).pack()
    Label(text="", width=10).pack()
    Label(text="", width=10).pack()

    main_screen.mainloop()

```

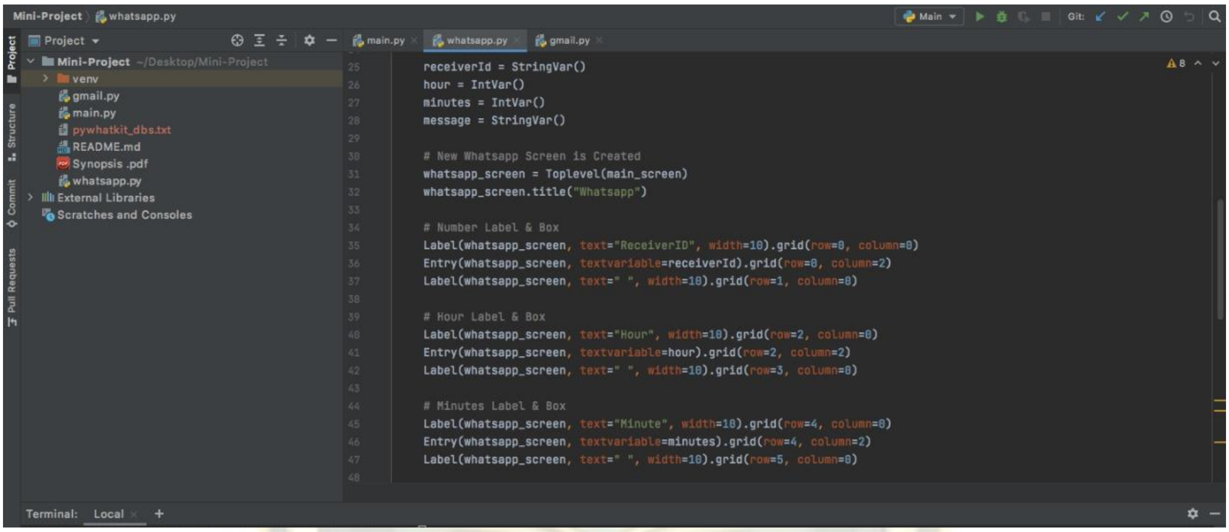
Whatsapp.py

1.



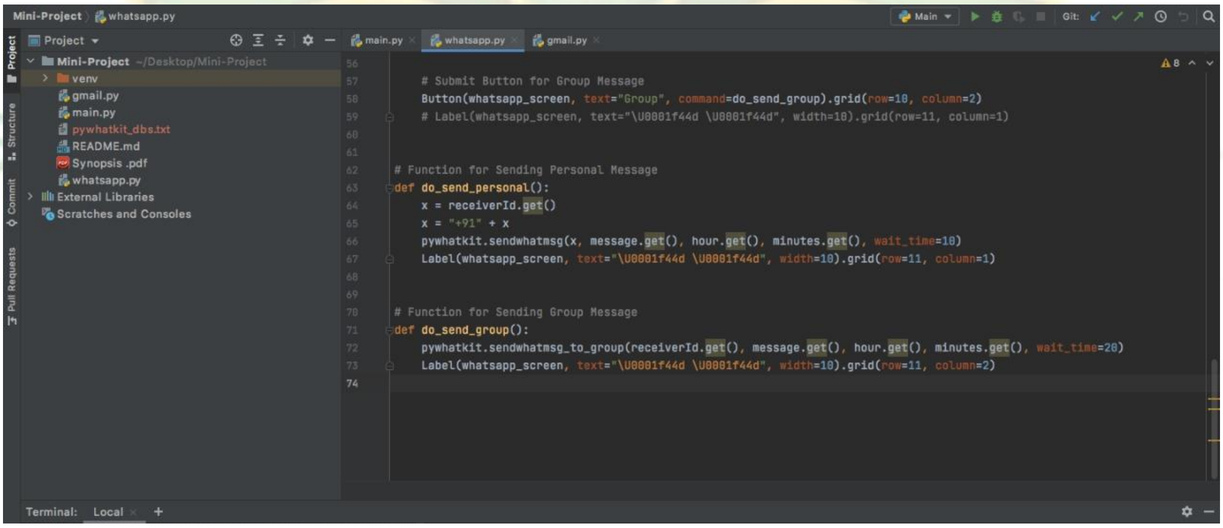
```python
# Import Statements
import pywhatkit
from tkinter import *
from tkinter.ttk import *

# Global Variable is Declared and Initialize as None
receiverId = None
hour = None
minutes = None
message = None
whatsapp_screen = None
main_screen = None


# Function for User Interface for Whatsapp Screen
def user_interface():
    # Global variable Assigned Datatype
    global main_screen
    global whatsapp_screen
    global receiverId
    global hour
    global minutes
    global message

```

2.



```python
    receiverId = StringVar()
    hour = IntVar()
    minutes = IntVar()
    message = StringVar()

    # New Whatsapp Screen is Created
    whatsapp_screen = Toplevel(main_screen)
    whatsapp_screen.title("Whatsapp")

    # Number Label & Box
    Label(whatsapp_screen, text="ReceiverID", width=10).grid(row=0, column=0)
    Entry(whatsapp_screen, textvariable=receiverId).grid(row=0, column=2)
    Label(whatsapp_screen, text=" ", width=10).grid(row=1, column=0)

    # Hour Label & Box
    Label(whatsapp_screen, text="Hour", width=10).grid(row=2, column=0)
    Entry(whatsapp_screen, textvariable=hour).grid(row=2, column=2)
    Label(whatsapp_screen, text=" ", width=10).grid(row=3, column=0)

    # Minutes Label & Box
    Label(whatsapp_screen, text="Minute", width=10).grid(row=4, column=0)
    Entry(whatsapp_screen, textvariable=minutes).grid(row=4, column=2)
    Label(whatsapp_screen, text=" ", width=10).grid(row=5, column=0)

```

3.



```python

    # Submit Button for Group Message
    Button(whatsapp_screen, text="Group", command=do_send_group).grid(row=10, column=2)
    # Label(whatsapp_screen, text="\U0001f44d \U0001f44d", width=10).grid(row=11, column=1)


# Function for Sending Personal Message
def do_send_personal():
    x = receiverId.get()
    x = "+91" + x
    pywhatkit.sendwhatmsg(x, message.get(), hour.get(), minutes.get(), wait_time=10)
    Label(whatsapp_screen, text="\U0001f44d \U0001f44d", width=10).grid(row=11, column=1)


# Function for Sending Group Message
def do_send_group():
    pywhatkit.sendwhatmsg_to_group(receiverId.get(), message.get(), hour.get(), minutes.get(), wait_time=20)
    Label(whatsapp_screen, text="\U0001f44d \U0001f44d", width=10).grid(row=11, column=2)

```

REFERENCES

- *https://www.w3schools.com/*
- *https://stackoverflow.com/*
- *https://www.tutorialspoint.com/*
- *https://www.youtube.com/*