R.V. COLLEGE OF ENGINEERING,
BENGALURU-560059
(Autonomous Institution Affiliated to VTU, Belagavi)

Python LAB EL

GUI Program - ProEditor

Python PLC Lab EL Report

Submitted By :-

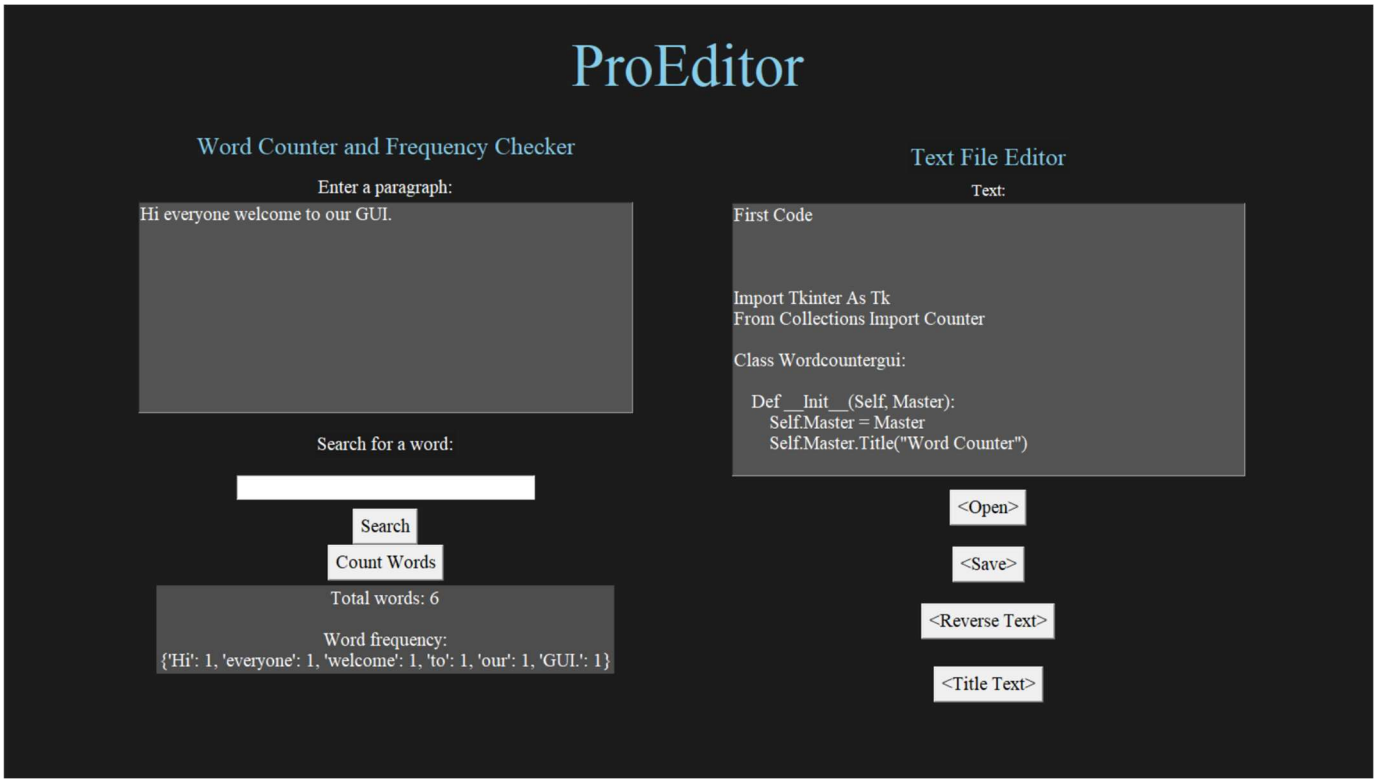
SNEHIL VUKKUSILA --- RVCE22BAI006

PARTH SHUKLA---RVCE22BAI031

Department of Artifial Intelligence And Machine Learning,
RV COLLEGE OF ENGINEERING®, BENGALURU – 560059

This code is a program that creates a GUI (graphical user interface) with two different programs. The first program is a word counter, in which the user can enter a paragraph or any text into the input box, and by utlising the button operations provided, he can press Count Words, this will count the total number of words entered, and also provide an ouput box beneath with a list containing the count of each specific word , and similarly provide a dialog box with this output as well. By entering a searchword into the input box, the user can also press the Search Word operation button, this will provide an output telling whether that specific word is present in the text or not.

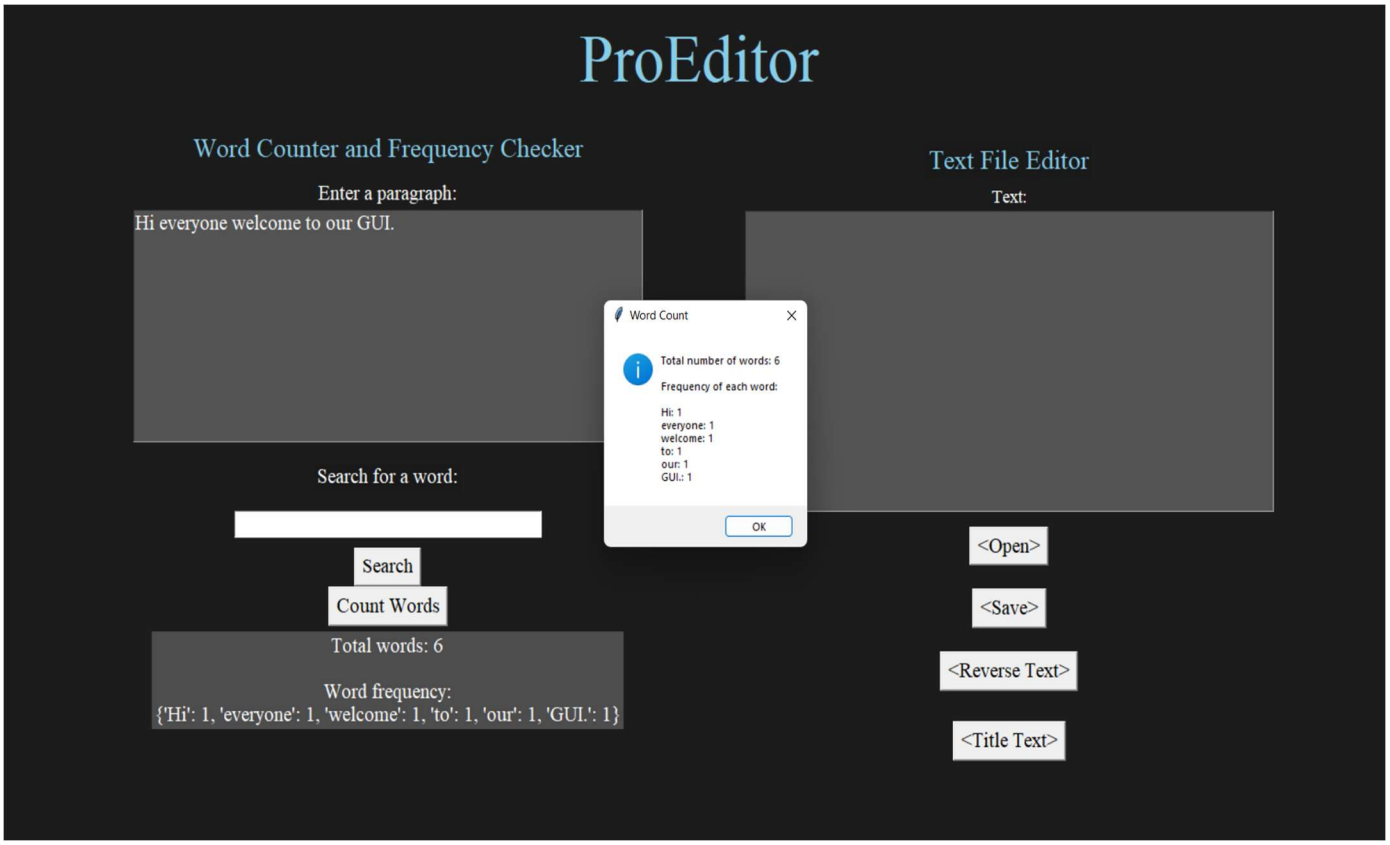
The second program is a text file editor, in which the user can make use of the Open operation button, which will provide an interactive file explorer dashboard, in which only text files will be present, on choosing a text file to open, the text in that file will be outputted in the text box above. The user can then make any changes he wants, add new text, remove text, change length, and make edits. Further the user can use the Save operation button to again be prompted with an interactive file explorer dashboard , the user can save a textfile of whatever name he chooses, or he may replace an existing textfile and override the save with the text he has entered in the text box provided in the GUI program.



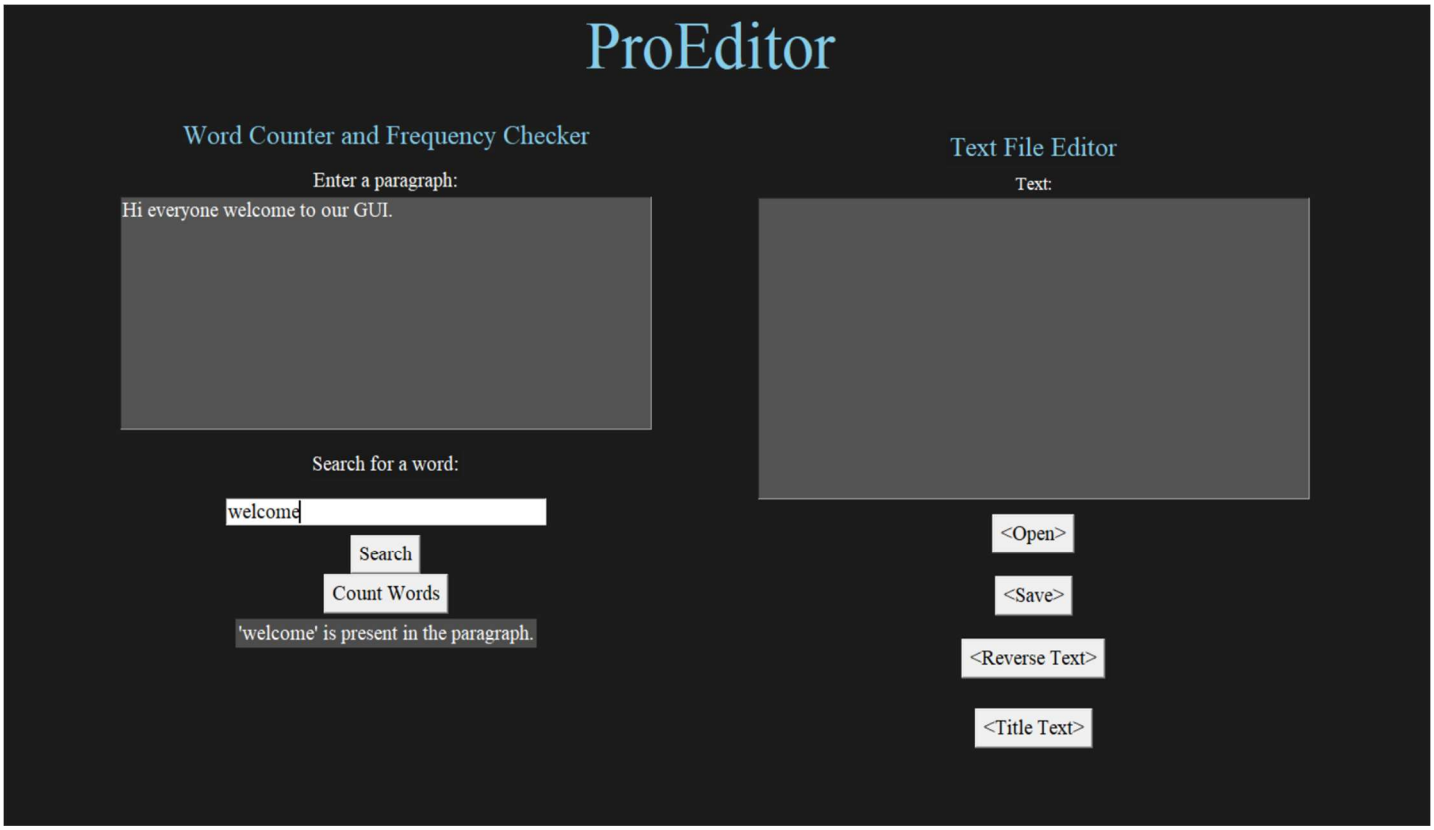
The GUI is using the TKINTER library and other modules related to TKINTER, such as frames, file accessing, boxes, Titles, Logos, RadioButtons, Interactive user entry boxes in which text can be entered to be operated upon, scrollable textbox when large text files are opened in the text editor box and many more features.

The program imports several modules from the tkinter library, including the main module, messagebox, and filedialog. The program also imports the ScrolledText module to create a scrolling text box.

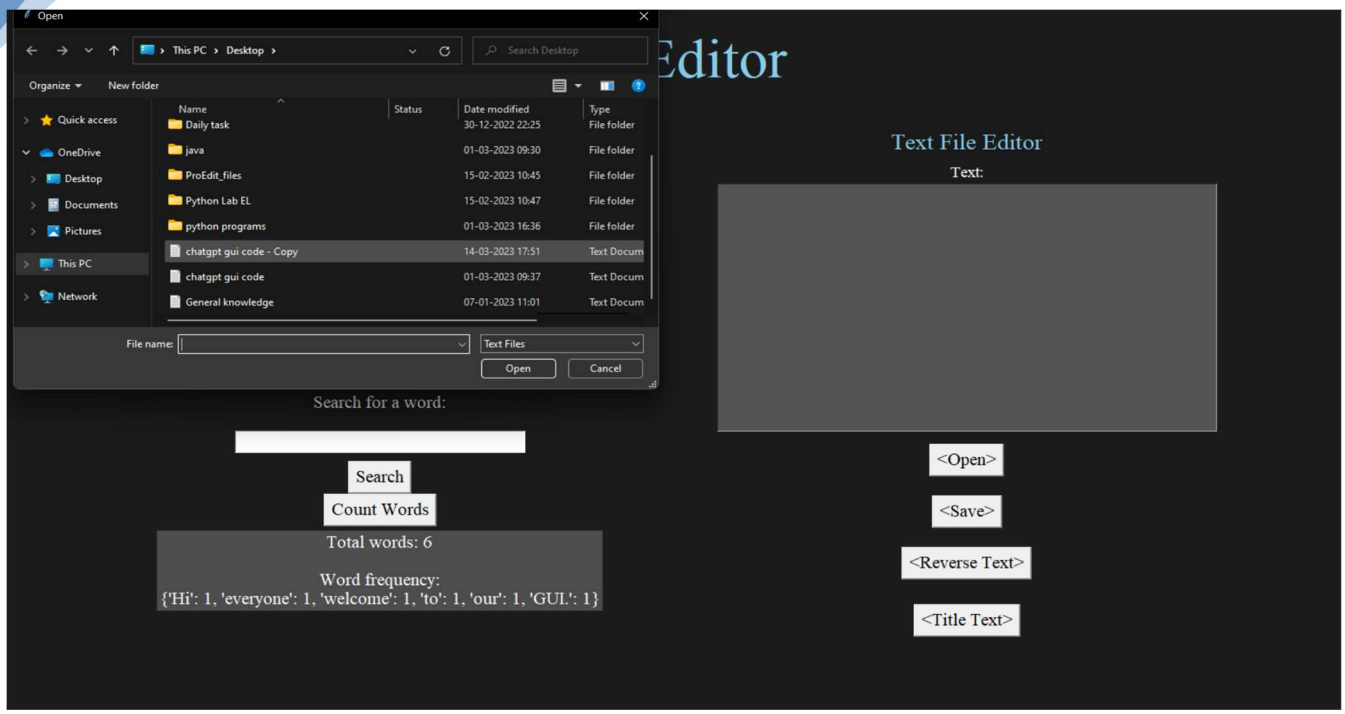
The GUI window is created with the title "ProEditor," and the attribute "-fullscreen" set to True, which means that the window will open in full screen. The canvas is created with a gradient background and is packed to fill the entire window.



The first program is located in the left frame of the canvas. It includes a title label, an input label, and a text box for users to enter a paragraph. A button labeled "Count Words" is also present to count the total number of words and frequency of each word in the paragraph.

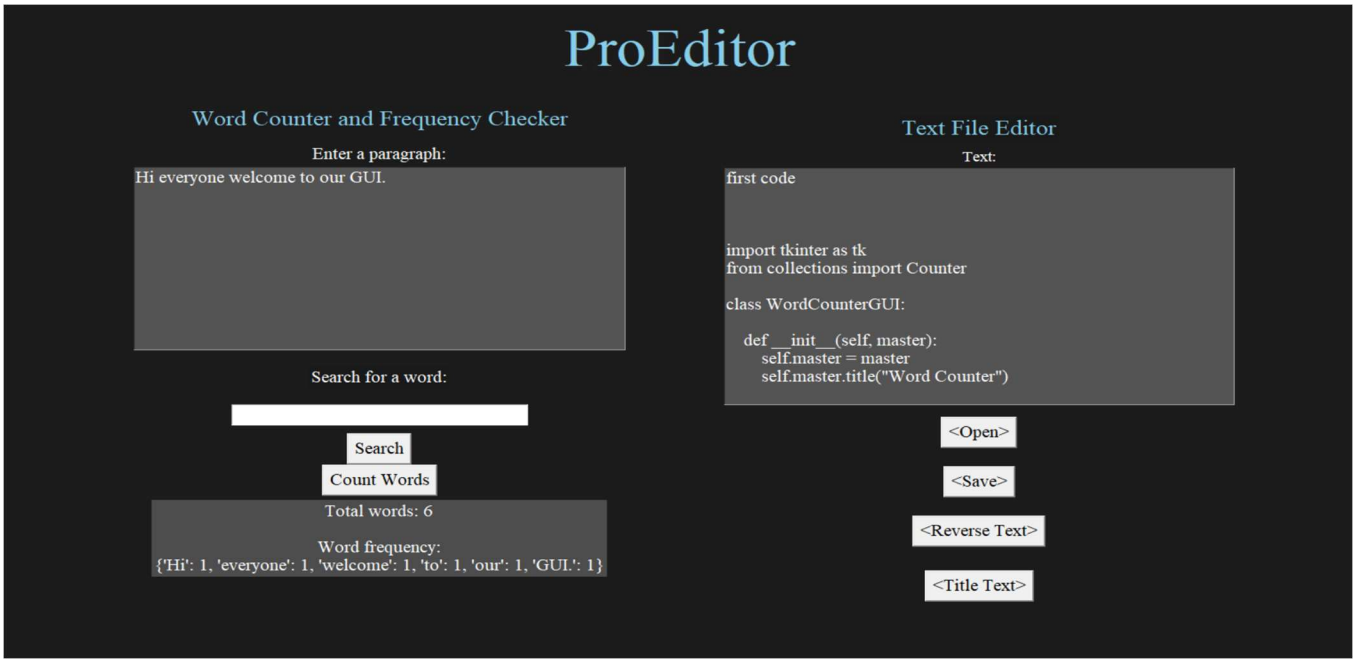


A search box and search button are also present to search for a specific word in the paragraph. An output label is included to display the results of the word count and frequency or search.



The second program is located in the right frame of the canvas. It includes a title label, an input label, and a text box for users to input or edit text. Buttons labeled "Open", "Save" , “Reverse Text” and "Title Text" are also present to perform actions on the text box.

Another scrolling text box is also present to display the contents of a selected file.Overall, the code creates a user-friendly interface for users to perform word counting, frequency checking, text editing, and file manipulation.



```
from tkinter import *
from tkinter import messagebox
from tkinter.filedialog import askopenfilename,
asksaveasfilename from tkinter.scrolledtext import
ScrolledText

# Creating the GUI window
root = Tk()
root.title("ProEditor")
root.attributes('-fullscreen', True)

# Creating a canvas with a gradient background
canvas = Canvas(root, bg="grey11",
highlightthickness=0) canvas.pack(fill=BOTH,
expand=YES)

# Creating the title label at the top center of the window
title_label = Label(canvas, text="ProEditor", font=("Times New
Roman", 50), fg="sky blue", bg="grey11")
title_label.place(relx=0.5, rely=0.08, anchor="center")

# Creating the left frame for the first program
left_frame = Frame(canvas, bg="grey11")
left_frame.place(relx=0.08, rely=0.15, relwidth=0.4,
relheight=1.0, anchor=NW)
left_label =Label(left_frame, text="Word Counter and
Frequency Checker", font=("Times New Roman", 20),
bg="grey11",
                        fg="sky blue")
left_label.pack(side="top", pady=10)

# Creating the input label and entry box for the first
program left_input_label = Label(left_frame, text="Enter
a paragraph:", font=("Times New Roman", 16), bg="grey11",
fg="white") left_input_label.pack(side="top", pady=2)
left_input_box = Text(left_frame, width=50, height=10,
font=("Times New Roman", 16), bg="grey33", fg="white")
left_input_box.pack(side="top", padx=20, pady=(0, 10))

ScrolledText(left_input_box,
                                    wrap = WORD,
                                    width = 40,
                                   height = 10,
                                    font = ("Times New Roman",
                                            20))
```

```
def count_words():
    # Get the text from the input box
    text = left_input_box.get("1.0", "end-1c")

    # Count the total number of words
    word_list = text.split()
    word_count = len(word_list)

    # Count the frequency of each word
    frequency_dict = {}
    for word in word_list:
        if word not in frequency_dict:
            frequency_dict[word] = 1
        else:
            frequency_dict[word] += 1
    output = f"Total words: {word_count}\n\nWord
frequency:\n{frequency_dict}"
    output_label.config(text=output)

    # Create a message box with the word count and
    frequency message = f'Total number of words:
    {word_count}\n\n' message += 'Frequency of each
    word:\n\n'
    for word, frequency in frequency_dict.items():
        message += f'{word}: {frequency}\n'
    messagebox.showinfo(title='Word Count', message=message)



# Creating the search label and entry box for the first
program
search_label = Label(left_frame, text="Search for a word:",
font=("Times New Roman", 16), fg="white", bg="#1a1a1a")
search_label.pack(side=TOP,pady=10)
search_entry = Entry(left_frame, width= 30,font=("Times New
Roman", 16))
search_entry.pack(side=TOP, padx=10, pady=10)
```

```
# Creating the search button for the first program
def search_word():
    paragraph = left_input_box.get("1.0", "end-1c")

    word = search_entry.get()
    if word in paragraph:
   output_label.config(text=f"'{word}' is present in the
paragraph.")
    else:
        output_label.config(text=f"'{word}' is not present
in the paragraph.")



search_button = Button(left_frame, text="Search",
font=("Times New Roman", 16), command=search_word)
search_button.pack()

count_button = Button(left_frame, text="Count
Words",font=("Times New Roman", 16), command=count_words)
count_button.pack()



# Creating the output label and text box for the first program
output_label = Label(left_frame, text="", font=("Times New
Roman", 16), fg="white", bg="gray30")
output_label.pack(side=TOP, pady=5)



#creating the right frame for the second program
right_frame = Frame(canvas, bg="grey11")
right_frame.place(relx=0.92, rely=0.17, relwidth=0.4,
relheight=1.0, anchor='ne')
right_label =Label(right_frame, text="Text File Editor",
font=("Times New Roman", 20), bg="#1a1a1a",
                        fg="#87ceeb")
right_label.pack(side="top", pady=5)
```

```
# Creating the input label and entry box for the first program
right_input_label = Label(right_frame, text="Text:",
font=("Times New Roman", 14), bg="grey11", fg="white")
right_input_label.pack(side="top", pady=0)
right_input_box = Text(right_frame, width=80, height=13,
font=("Times New Roman", 16), bg="grey33", fg="white")
right_input_box.pack(side="top", padx=20, pady=0)

ScrolledText(right_input_box,
                                    wrap = WORD,
                                    width = 26,
                                    height = 10,
                                    font = ("Times New Roman",
                                            20))



def open_file():
    """Open a file for editing."""
    filepath = askopenfilename(
        filetypes=[("Text Files", "*.txt"), ("All Files",
    "*.*")] )
    if not filepath:
        return
    right_input_box.delete("1.0", "end-1c")
    with open(filepath, "r") as input_file:
        text = input_file.read()
        right_input_box.insert("end-1c", text)
    canvas.title(f"Text Editor Application - {filepath}")


def save_file():
    """Save the current file as a new file."""
    filepath = asksaveasfilename(
        defaultextension="txt",
        filetypes=[("Text Files", "*.txt"), ("All Files",
    "*.*")], )
    if not filepath:
        return
    with open(filepath, "w") as output_file:
        text = right_input_box.get("1.0", "end-1c")
        output_file.write(text)
 right_frame.title(f"Text Editor Application - {filepath}")


btn_open = Button(right_frame, text="<Open>", font=("Times New
```

```
Roman", 16), command=open_file)
btn_open.pack(pady=15)
btn_save = Button(right_frame, text="<Save>", font=("Times New
Roman", 16), command=save_file)
btn_save.pack(pady=8)


# Start the main loop
root.mainloop()
```