VAULT OF CODES

To-do-list application

Python Internship Major Project

Major Project:- **Write a code of To-Do-List Application GUI by using python.**

Source Code:-

```
# importing the required modules
from tkinter import *
# importing the messagebox module from the tkinter library
from tkinter import messagebox
# importing the sqlite3 module as sql
import sqlite3 as sql


# defining the function to add tasks to the list
def add_task():
    # getting the string from the entry field
    task_string = task_field.get()
    # checking whether the string is empty or not
    if len(task_string) == 0:
```

```
        # displaying a message box with 'Empty Field' message
        messagebox.showinfo('Error', 'Field is Empty.')
    else:
        # adding the string to the tasks list
        tasks.append(task_string)
        # using the execute() method to execute a SQL statement
        the_cursor.execute('insert into tasks values (?)', (task_string ,))
        # calling the function to update the list
        list_update()
        # deleting the entry in the entry field
        task_field.delete(0, 'end')

# defining the function to update the list
def list_update():
    # calling the function to clear the list
    clear_list()
    # iterating through the strings in the list
    for task in tasks:
        # using the insert() method to insert the tasks in the list box
        task_listbox.insert('end', task)

# defining the function to delete a task from the list
def delete_task():
    # using the try-except method
    try:
        # getting the selected entry from the list box
        the_value = task_listbox.get(task_listbox.curselection())
        # checking if the stored value is present in the tasks list
        if the_value in tasks:
```

```
        # removing the task from the list
        tasks.remove(the_value)
        # calling the function to update the list
        list_update()
        # using the execute() method to execute a SQL statement
        the_cursor.execute('delete from tasks where title = ?', (the_value,))
    except:
        # displaying the message box with 'No Item Selected' message for an exception
        messagebox.showinfo('Error', 'No Task Selected. Cannot Delete.')

# function to delete all tasks from the list
def delete_all_tasks():
    # displaying a message box to ask user for confirmation
    message_box = messagebox.askyesno('Delete All', 'Are you sure?')
    # if the value turns to be True
    if message_box == True:
        # using while loop to iterate through the tasks list until it's empty
        while(len(tasks) != 0):
            # using the pop() method to pop out the elements from the list
            tasks.pop()
        # using the execute() method to execute a SQL statement
        the_cursor.execute('delete from tasks')
        # calling the function to update the list
        list_update()

# function to clear the list
def clear_list():
    # using the delete method to delete all entries from the list box
    task_listbox.delete(0, 'end')
```

```
# function to close the application
def close():
    # printing the elements from the tasks list
    print(tasks)
    # using the destroy() method to close the application
    guiWindow.destroy()

# function to retrieve data from the database
def retrieve_database():
    # using the while loop to iterate through the elements in the tasks list
    while(len(tasks) != 0):
        # using the pop() method to pop out the elements from the list
        tasks.pop()
    # iterating through the rows in the database table
    for row in the_cursor.execute('select title from tasks'):
        # using the append() method to insert the titles from the table in the list
        tasks.append(row[0])

# main function
if __name__ == "__main__":
    # creating an object of the Tk() class
    guiWindow = Tk()
    # setting the title of the window
    guiWindow.title("To-Do List ")
    # setting the geometry of the window
    guiWindow.geometry("665x400+550+250")
    # disabling the resizable option
    guiWindow.resizable(0, 0)
```

```
# setting the background color to #B5E5CF
guiWindow.configure(bg = "#B5E5CF")

# using the connect() method to connect to the database
the_connection = sql.connect('listOfTasks.db')
# creating the cursor object of the cursor class
the_cursor = the_connection.cursor()
# using the execute() method to execute a SQL statement
the_cursor.execute('create table if not exists tasks (title text)')

# defining an empty list
tasks = []

# defining frames using the tk.Frame() widget
functions_frame = Frame(guiWindow, bg = "black")

# using the pack() method to place the frames in the application
functions_frame.pack(side = "top", expand = True, fill = "both")



# defining another label using the Label() widget
task_label = Label( functions_frame,text = "Enter the Task:",
    font = ("arial", "14", "bold"),
    background = "black",
    foreground="white"
)
# using the place() method to place the label in the application
task_label.place(x = 20, y = 30)
```

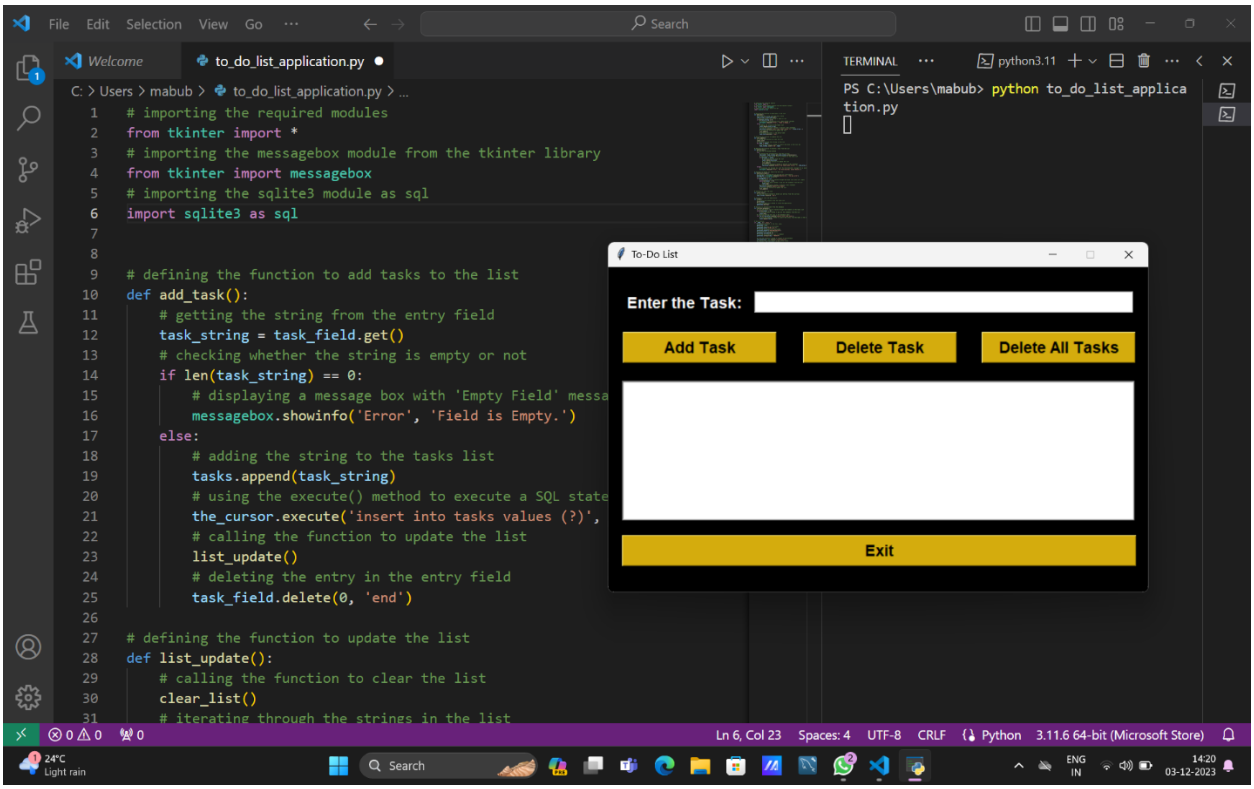
```
# defining an entry field using the Entry() widget
task_field = Entry(
    functions_frame,
    font = ("Arial", "14"),
    width = 42,
    foreground="black",
    background = "white",
)
# using the place() method to place the entry field in the application
task_field.place(x = 180, y = 30)

# adding buttons to the application using the Button() widget
add_button =Button(
    functions_frame,
    text = "Add Task",
    width = 15,
    bg='#D4AC0D',font=("arial", "14", "bold"),
    command = add_task,

)
del_button = Button(
    functions_frame,
    text = "Delete Task",
    width = 15,
    bg='#D4AC0D', font=("arial", "14", "bold"),
    command = delete_task,
)
del_all_button = Button(
```

```
    functions_frame,
    text = "Delete All Tasks",
    width = 15,
    font=("arial", "14", "bold"),
    bg='#D4AC0D',
    command = delete_all_tasks
)
exit_button = Button(
    functions_frame,
    text = "Exit",
    width = 52,
    bg='#D4AC0D',  font=("arial", "14", "bold"),
    command = close
)
# using the place() method to set the position of the buttons in the application
add_button.place(x = 18, y = 80,)
del_button.place(x = 240, y = 80)
del_all_button.place(x = 460, y = 80)
exit_button.place(x = 17, y = 330)

# defining a list box using the tk.Listbox() widget
task_listbox = Listbox(
    functions_frame,
    width = 57,
    height = 7,
    font="bold",
    selectmode = 'SINGLE',
    background = "WHITE",
    foreground="BLACK",
```

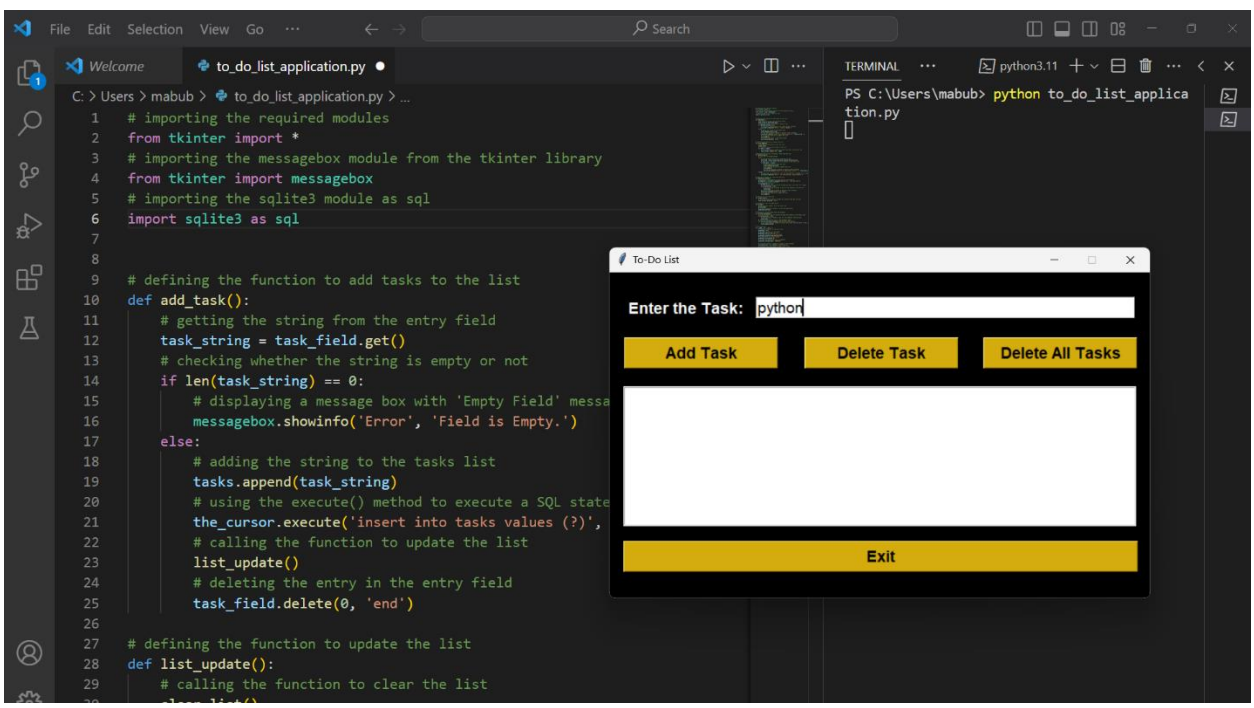
```
    selectbackground = "#D4AC0D",
    selectforeground="BLACK"
)
# using the place() method to place the list box in the application
task_listbox.place(x = 17, y = 140)

# calling some functions
retrieve_database()
list_update()
# using the mainloop() method to run the application
guiWindow.mainloop()
# establishing the connection with database
the_connection.commit()
the_cursor.close()
```

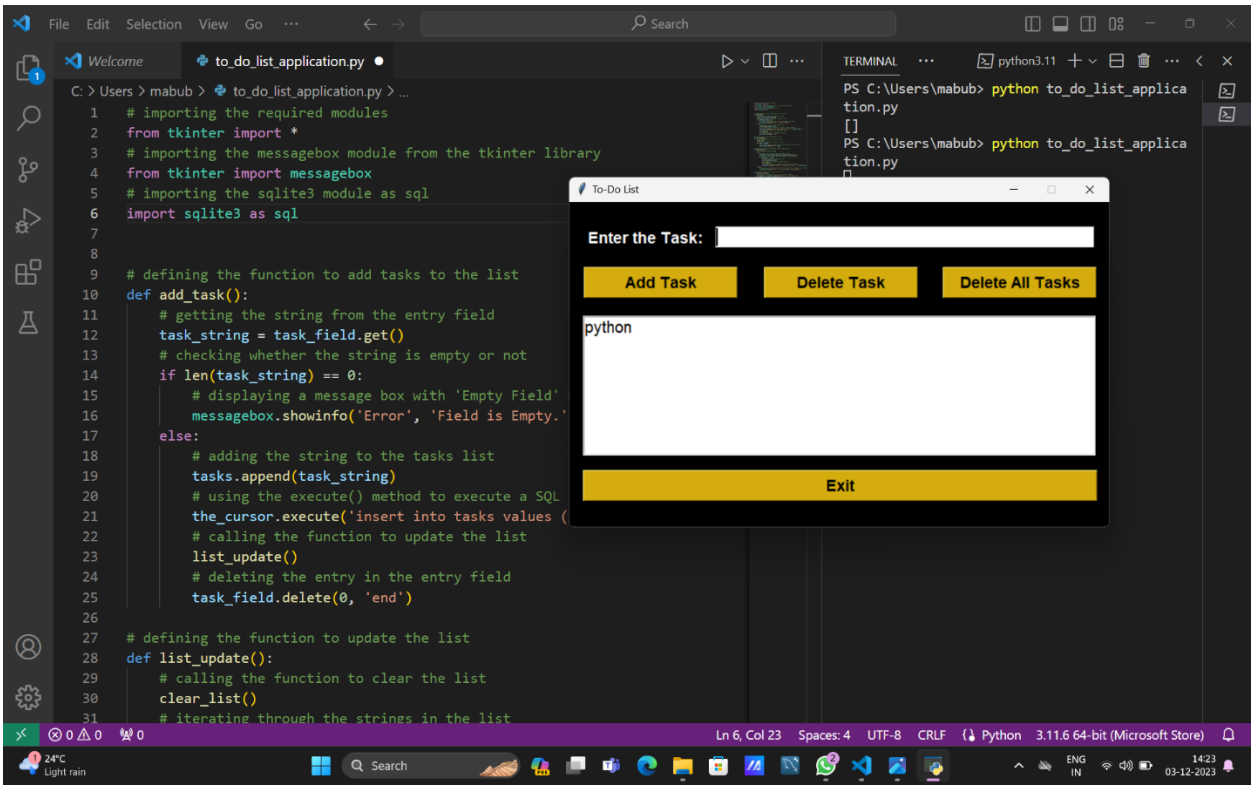
OUTPUT:-

- Run the Above code we can see like this….

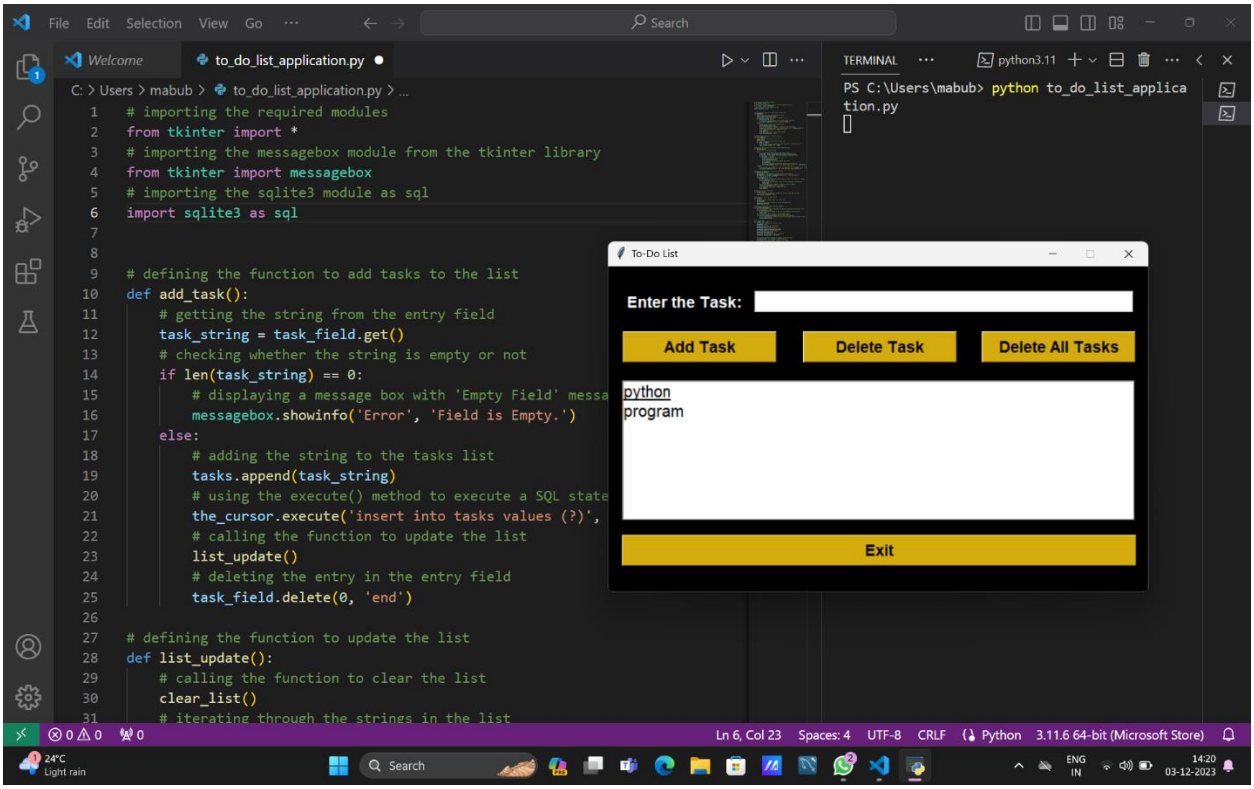


- After we can enter the task and click on the “Add Task”

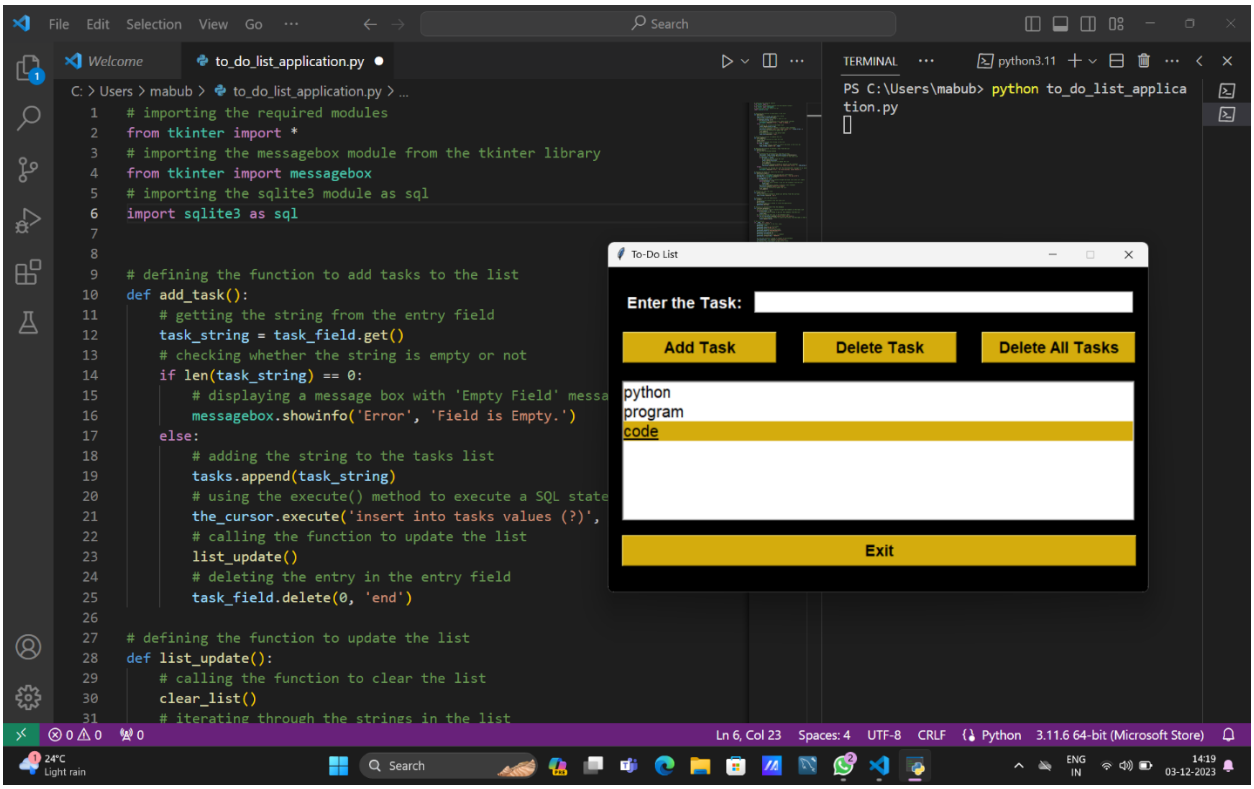


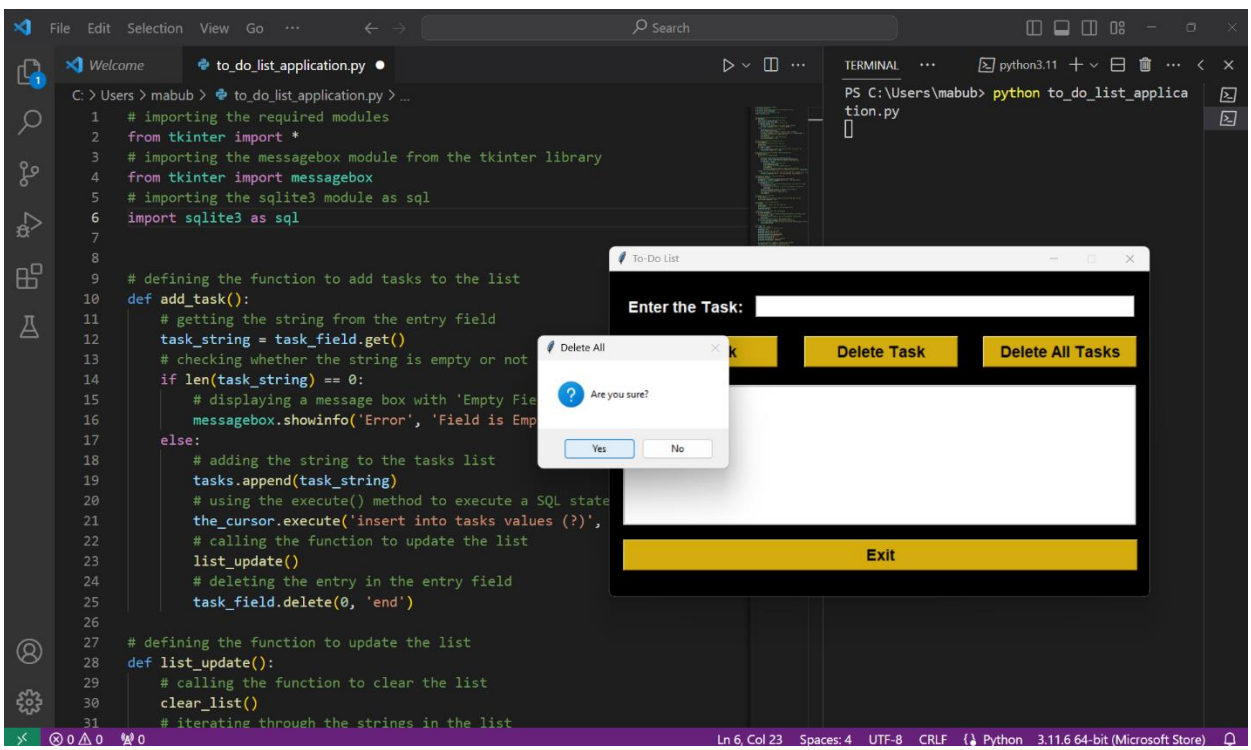


- Like this we can enter the task more…and click on the Add Task

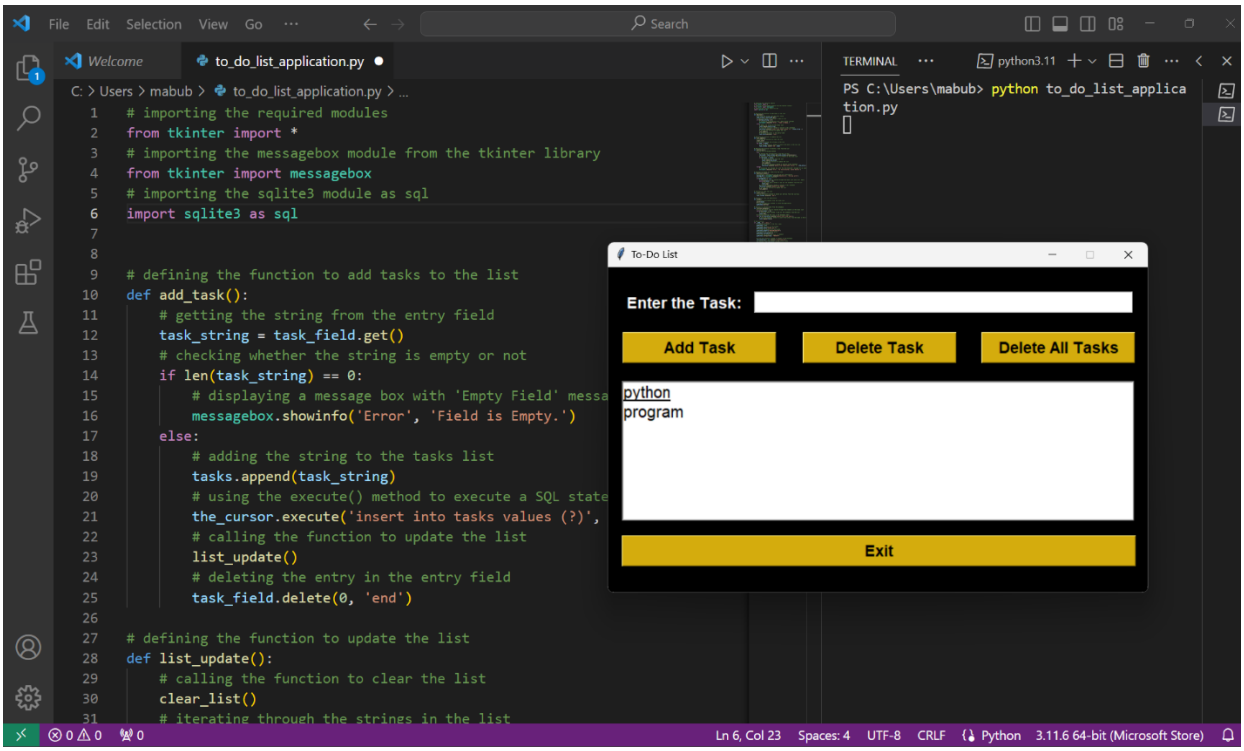


- Now, we want to delete the task ,firstly select the task and after that click on the yes .

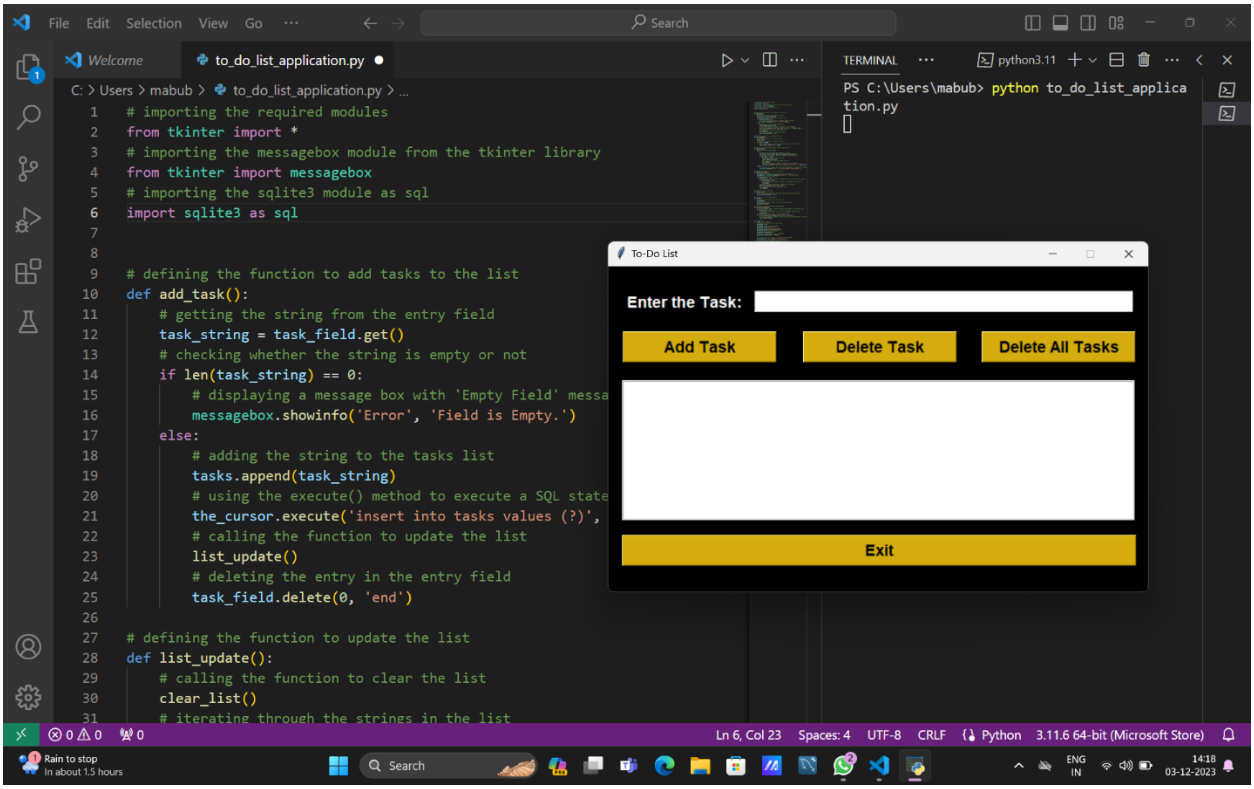




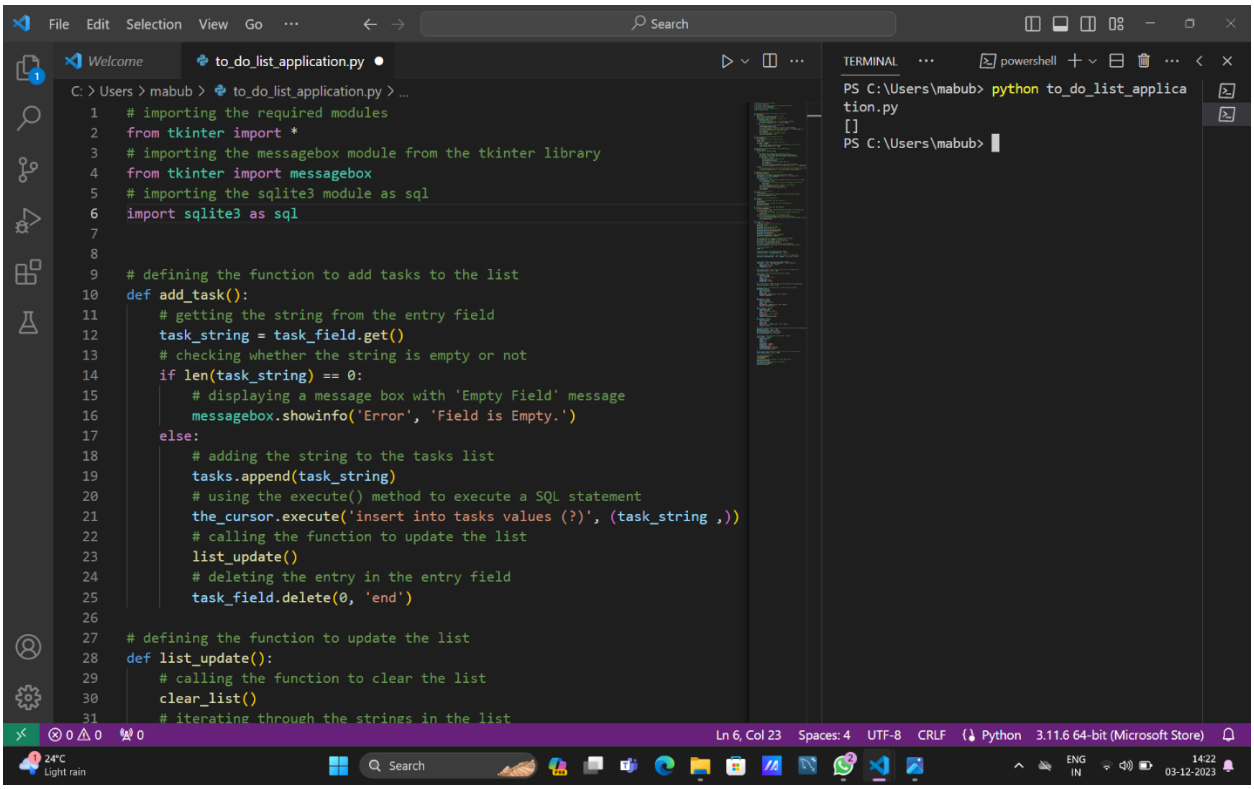
- After that we can see like this…….



- Now,we want to delete All Tasks…click on the Delete all tasks.



- After that, click on the Exit back to the terminal.



THANK YOU

Submitted by

B.CHAITANYA