

```

CODE:
from tkinter import *
from tkinter.ttk import *
from tkinter import font,colorchooser,filedialog,messagebox
import os
import tempfile
from datetime import datetime

#Funtionality Part

def date_time(event=None):
    currentdatetime=datetime.now() #20/06/2022, May 20, 2022 06/20/22
    sept-20-22
    formatteddatetime=currentdatetime.strftime('%b-%d-%y %H:%M:%S')
    textarea.insert(1.0,formatteddatetime)

def printout(event=None):
    file=tempfile.mktemp('.txt')
    open(file,'w').write(textarea.get(1.0,END))
    os.startfile(file,'print')

def change_theme(bg_color,fg_color):
    textarea.config(bg=bg_color,fg=fg_color)

def toolbarFunc():
    if show_toolbar.get()==False:
        tool_bar.pack_forget()
    if show_toolbar.get()==True:
        textarea.pack_forget()
        tool_bar.pack(fill=X)
        textarea.pack(fill=BOTH,expand=1)

def statusbarFunc():
    if show_statusbar.get()==False:
        status_bar.pack_forget()
    else:
        status_bar.pack()

def find():
    #funtionality

    def find_words():
        textarea.tag_remove('match',1.0,END)
        start_pos='1.0'
        word=findentryField.get()
        if word:
            while True: #I am a boy I am 16 yrs old
                start_pos=textarea.search(word,start_pos,stopindex=END) #1.0

```

```

        if not start_pos:
            break
        end_pos=f'{start_pos}+{len(word)}c' #1.0+1c
        textarea.tag_add('match',start_pos,end_pos)

textarea.tag_config('match',foreground='red',background='yellow')
start_pos=end_pos

def replace_text():
    word = findentryField.get()
    replaceword=replaceentryField.get()
    content=textarea.get(1.0,END)
    new_content=content.replace(word,replaceword)
    textarea.delete(1.0,END)
    textarea.insert(1.0,new_content)

#GUI
root1=Toplevel()

root1.title('Find')
root1.geometry('450x250+500+200')
root1.resizable(0,0)

labelFrame=LabelFrame(root1,text='Find/Replace')
labelFrame.pack(pady=50)

findLabel=Label(labelFrame,text='Find')
findLabel.grid(row=0,column=0,padx=5,pady=5)
findentryField=Entry(labelFrame)
findentryField.grid(row=0,column=1,padx=5,pady=5)

replaceLabel = Label(labelFrame, text='Replace')
replaceLabel.grid(row=1, column=0, padx=5, pady=5)
replaceentryField = Entry(labelFrame)
replaceentryField.grid(row=1, column=1, padx=5, pady=5)

findButton=Button(labelFrame,text='FIND',command=find_words)
findButton.grid(row=2,column=0,padx=5,pady=5)

replaceButton = Button(labelFrame,
text='REPLACE',command=replace_text)
replaceButton.grid(row=2, column=1, padx=5, pady=5)

def doSomething():
    textarea.tag_remove('match',1.0,END)
    root1.destroy()

root1.protocol('WM_DELETE_WINDOW',doSomething)
root1.mainloop()

```

```

def statusBarFunction(event):
    # if textarea.edit_modified():
        words=len(textarea.get(0.0,END).split())
        characters=len(textarea.get(0.0,'end-1c').replace(' ',''))
#1.0
    status_bar.config(text=f'Characters: {characters} Words:
{words}')

    #
        textarea.edit_modified(False)

url=''
def new_file(event=None):
    global url
    url=''
    textarea.delete(0.0,END)

def open_file(event=None):
    global url
    url=filedialog.askopenfilename(initialdir=os.getcwd,title='Select
File',filetypes=(('Text File','txt'),
                    ('All Files','*.*')))
    if url != '':
        data=open(url,'r')
        textarea.insert(0.0,data.read())
        root.title(os.path.basename(url))

def save_file(event=None):
    if url == '':

save_url=filedialog.asksaveasfile(mode='w',defaultextension='.txt',fil
etypes=(('Text File','txt'),
        ('All Files','*.*')))
    if save_url is None:
        pass
    else:
        content=textarea.get(0.0,END)
        save_url.write(content)
        save_url.close()

    else:
        content=textarea.get(0.0,END)
        file=open(url,'w')
        file.write(content)

def saveas_file(event=None):
    save_url = filedialog.asksaveasfile(mode='w',
defaultextension='.txt', filetypes=(('Text File', 'txt'),
        ('All Files', '*.*')))

```

```

        content = textarea.get(0.0, END)
        save_url.write(content)
        save_url.close()
        if url != '':
            os.remove(url)

def iexit(event=None):
    if textarea.edit_modified():
        result=messagebox.askyesnocancel('Warning','Do you want to
save the file?')
        if result is True:
            if url!='':
                content=textarea.get(0.0,END)
                file=open(url,'w')
                file.write(content)
                root.destroy()
            else:
                content=textarea.get(0.0,END)
                save_url = filedialog.asksaveasfile(mode='w',
defaulttextension='.txt', filetypes=(('Text File', 'txt'),
('All Files', '*.*')))

                save_url.write(content)
                save_url.close()
                root.destroy()

        elif result is False:
            root.destroy()

        else:
            pass

    else:
        root.destroy()

fontSize=12
fontStyle='arial'
def font_style(event):
    global fontStyle
    fontStyle=font_family_variable.get()
    textarea.config(font=(fontStyle,fontSize))

def font_size(event):
    global fontSize
    fontSize=size_variable.get()
    textarea.config(font=(fontStyle,fontSize))

def bold_text():
    text_property=font.Font(font=textarea['font']).actual()
    if text_property['weight']=='normal':

```

```

        textarea.config(font=(fontStyle,fontSize,'bold'))

    if text_property['weight']=='bold':
        textarea.config(font=(fontStyle,fontSize,'normal'))

def italic_text():
    text_property = font.Font(font=textarea['font']).actual()
    if text_property['slant']=='roman':
        textarea.config(font=(fontStyle,fontSize,'italic'))

    if text_property['slant']=='italic':
        textarea.config(font=(fontStyle,fontSize,'roman'))

def underline_text():
    text_property = font.Font(font=textarea['font']).actual()
    if text_property['underline']==0:
        textarea.config(font=(fontStyle,fontSize,'underline'))

    if text_property['underline']==1:
        textarea.config(font=(fontStyle,fontSize,))

def color_select():
    color=colorchooser.askcolor()
    textarea.config(fg=color[1])

def align_right():
    data=textarea.get(0.0,END)
    textarea.tag_config('right',justify=RIGHT)
    textarea.delete(0.0,END)
    textarea.insert(INSERT,data,'right')

def align_left():
    data=textarea.get(0.0,END)
    textarea.tag_config('left',justify=LEFT)
    textarea.delete(0.0,END)
    textarea.insert(INSERT,data,'left')

def align_center():
    data=textarea.get(0.0,END)
    textarea.tag_config('center',justify=CENTER)
    textarea.delete(0.0,END)
    textarea.insert(INSERT,data,'center')

root=Tk()
root.title('Text Editor')
root.geometry('1200x620+10+10')
root.resizable(False,False)
menubar=Menu(root)
root.config(menu=menubar)

#file menu section

```

```

filemenu=Menu(menubar,tearoff=False)
menubar.add_cascade(label='File',menu=filemenu)
newImage=PhotoImage(file='new.png')
openImage=PhotoImage(file='open.png')
saveImage=PhotoImage(file='save.png')
saveasImage=PhotoImage(file='save_as.png')
exitImage=PhotoImage(file='exit.png')
printImage=PhotoImage(file='print.png')
filemenu.add_command(label='New',accelerator='Ctrl+N',image=newImage,c
ompound=LEFT,command=new_file)
filemenu.add_command(label='Open',accelerator='Ctrl+O',image=openImage
,compound=LEFT,command=open_file)
filemenu.add_command(label='Save',accelerator='Ctrl+S',image=saveImage
,compound=LEFT,command=save_file)
filemenu.add_command(label='Save
As',accelerator='Ctrl+Alt+S',image=saveasImage,compound=LEFT,command=s
aveas_file)
filemenu.add_command(label='Print',accelerator='Ctrl+P',image=printIma
ge,compound=LEFT,command=printout)

filemenu.add_separator()
filemenu.add_command(label='Exit',accelerator='Ctrl+Q',image=exitImage
,compound=LEFT,command=iexit)

#Edit menu section
cutImage=PhotoImage(file='cut.png')
copyImage=PhotoImage(file='copy.png')
pasteImage=PhotoImage(file='paste.png')
clearImage=PhotoImage(file='clear_all.png')
findImage=PhotoImage(file='find.png')

#toolbar section

tool_bar=Label(root)
tool_bar.pack(side=TOP,fill=X)
font_families=font.families()
font_family_variable=StringVar()
fontfamily_Combobox=Combobox(tool_bar,width=30,values=font_families,st
ate='readonly',textvariable=font_family_variable)
fontfamily_Combobox.current(font_families.index('Arial'))
fontfamily_Combobox.grid(row=0,column=0,padx=5)
size_variable=IntVar()
font_size_Combobox=Combobox(tool_bar,width=14,textvariable=size_variab
le,state='readonly',values=tuple(range(8,81)))
font_size_Combobox.current(4)
font_size_Combobox.grid(row=0,column=1,padx=5)

fontfamily_Combobox.bind('<<ComboboxSelected>>','font_style')
font_size_Combobox.bind('<<ComboboxSelected>>','font_size')

#buttons section
boldImage=PhotoImage(file='bold.png')
boldButton=Button(tool_bar,image=boldImage,command=bold_text)
boldButton.grid(row=0,column=2,padx=5)

```

```

italicImage=PhotoImage(file='italic.png')
italicButton=Button(tool_bar,image=italicImage,command=italic_text)
italicButton.grid(row=0,column=3,padx=5)

underlineImage=PhotoImage(file='underline.png')
underlineButton=Button(tool_bar,image=underlineImage,command=underline_text)
underlineButton.grid(row=0,column=4,padx=5)

fontColorImage=PhotoImage(file='font_color.png')
fontColorButton=Button(tool_bar,image=fontColorImage,command=color_select)
fontColorButton.grid(row=0,column=5,padx=5)

leftAlignImage=PhotoImage(file='left.png')
leftAlignButton=Button(tool_bar,image=leftAlignImage,command=align_left)
leftAlignButton.grid(row=0,column=6,padx=5)

rightAlignImage=PhotoImage(file='right.png')
rightAlignButton=Button(tool_bar,image=rightAlignImage,command=align_right)
rightAlignButton.grid(row=0,column=7,padx=5)

centerAlignImage=PhotoImage(file='center.png')
centerAlignButton=Button(tool_bar,image=centerAlignImage,command=align_center)
centerAlignButton.grid(row=0,column=8,padx=5)

scrollbar=Scrollbar(root)
scrollbar.pack(side=RIGHT,fill=Y)
textarea=Text(root,yscrollcommand=scrollbar.set,font=('arial',12),undo=True)
textarea.pack(fill=BOTH,expand=True)
scrollbar.config(command=textarea.yview)

status_bar=Label(root,text='Status Bar')
status_bar.pack(side=BOTTOM)

textarea.bind('<<Modified>>',statusBarFunction)
selectImage=PhotoImage(file='checked.png')
undoImage=PhotoImage(file='undo.png')
datetimeImage=PhotoImage(file='calender.png')
editmenu=Menu(menu_bar,tearoff=False)
editmenu.add_command(label='Undo',accelerator='Ctrl+Z',image=undoImage,compound=LEFT,
)
editmenu.add_command(label='Cut',accelerator='Ctrl+X',image=cutImage,compound=LEFT,
command=lambda :textarea.event_generate('<Control x>'))
editmenu.add_command(label='Copy',accelerator='Ctrl+C',image=copyImage

```

```

,compound=LEFT,
                                command=lambda :textarea.event_generate('<Control
c>'))
editmenu.add_command(label='Paste', accelerator='Ctrl+V', image=pasteImage,
compound=LEFT,
                                command=lambda :textarea.event_generate('<Control
v>'))

editmenu.add_command(label='Select
All', accelerator='Ctrl+A', image=selectImage, compound=LEFT,
)

editmenu.add_command(label='Clear', accelerator='Ctrl+Alt+X', image=clearImage,
compound=LEFT,
                                command=lambda :textarea.delete(0.0,END))
editmenu.add_command(label='Find', accelerator='Ctrl+F', image=findImage,
,compound=LEFT,command=find)
editmenu.add_command(label='Time/Date', accelerator='Ctrl+D', image=date
timeImage, compound=LEFT
                                ,command=date_time)
menubar.add_cascade(label='Edit', menu=editmenu)

#view menu section
show_toolbar=BooleanVar()
show_statusbar=BooleanVar()
statusImage=PhotoImage(file='status_bar.png')
toolbarImage=PhotoImage(file='tool_bar.png')
viewmenu=Menu(menubar, tearoff=False)
viewmenu.add_checkbutton(label='Tool
Bar', variable=show_toolbar, onvalue=True, offvalue=False, image=toolbarImage,
                                compound=LEFT, command=toolbarFunc)
show_toolbar.set(True)
viewmenu.add_checkbutton(label='Status
Bar', variable=show_statusbar, onvalue=True, offvalue=False, image=statusImage,
                                compound=LEFT, command=statusbarFunc)
show_statusbar.set(True)
menubar.add_cascade(label='View', menu=viewmenu)

#themes menu section
themesmenu=Menu(menubar, tearoff=False)
menubar.add_cascade(label='Themes', menu=themesmenu)
theme_choice=StringVar()
lightImage=PhotoImage(file='light_default.png')
darkImage=PhotoImage(file='dark.png')
pinkImage=PhotoImage(file='red.png')
monokaiImage=PhotoImage(file='monokai.png')
themesmenu.add_radiobutton(label='Light
Default', image=lightImage, variable=theme_choice, compound=LEFT
                                ,command=lambda
:change_theme('white','black'))
themesmenu.add_radiobutton(label='Dark', image=darkImage, variable=theme
_choice, compound=LEFT

```



```

        ,command=lambda
:change_theme('gray20','white'))
themesmenu.add_radiobutton(label='Pink',image=pinkImage,variable=theme
_choice,compound=LEFT
        ,command=lambda
:change_theme('pink','blue'))
themesmenu.add_radiobutton(label='Monokai',image=monokaiImage,variable
=theme_choice,compound=LEFT
        ,command=lambda
:change_theme('orange','white'))

root.bind("<Control-o>",open_file)
root.bind("<Control-n>",new_file)
root.bind("<Control-s>",save_file)
root.bind("<Control-Alt-s>",saveas_file)
root.bind("<Control-q>",iexit)
root.bind("<Control-p>",printout)
root.bind("<Control-d>",date_time)

root.mainloop()

```