Desktop Application Development

1. **Tkinter Initialization:**
   * The TextEditor class initializes the Tkinter window with a title, geometry, and a Text widget to allow text editing.
2. **Menu Bar:**
   * A Menu bar is created with File and Edit menus.
   * The File menu contains options for creating a new file, opening an existing file, saving the current file, saving as a new file, and exiting the editor.
   * The Edit menu provides undo and redo functionalities.
3. **File Operations:**
   * new\_file: Clears the text area to start a new document.
   * open\_file: Opens a file dialog to choose a text file, reads its content, and displays it in the text area.
   * save\_file: Saves the content of the text area to a specified file.
   * save\_as\_file: Works like save\_file but always prompts for a file name.
   * exit\_editor: Prompts the user to confirm before exiting the application.
4. **Running the Application:**
   * The application is started by creating an instance of TextEditor and entering the main loop with root.mainloop().

This simple text editor can be expanded with more features like find/replace, formatting options, etc.

Source Code:

import tkinter as tk

from tkinter import filedialog

from tkinter import messagebox

class TextEditor:

def \_\_init\_\_(self, root):

self.root = root

self.root.title("Simple Text Editor")

self.root.geometry("600x400")

# Create a Text widget

self.text\_area = tk.Text(self.root, undo=True)

self.text\_area.pack(fill=tk.BOTH, expand=1)

# Create a Menu bar

self.menu\_bar = tk.Menu(self.root)

self.root.config(menu=self.menu\_bar)

# Add File menu

self.file\_menu = tk.Menu(self.menu\_bar, tearoff=0)

self.menu\_bar.add\_cascade(label="File", menu=self.file\_menu)

self.file\_menu.add\_command(label="New", command=self.new\_file)

self.file\_menu.add\_command(label="Open", command=self.open\_file)

self.file\_menu.add\_command(label="Save", command=self.save\_file)

self.file\_menu.add\_command(label="Save As", command=self.save\_as\_file)

self.file\_menu.add\_separator()

self.file\_menu.add\_command(label="Exit", command=self.exit\_editor)

# Add Edit menu

self.edit\_menu = tk.Menu(self.menu\_bar, tearoff=0)

self.menu\_bar.add\_cascade(label="Edit", menu=self.edit\_menu)

self.edit\_menu.add\_command(label="Undo", command=self.text\_area.edit\_undo)

self.edit\_menu.add\_command(label="Redo", command=self.text\_area.edit\_redo)

def new\_file(self):

self.text\_area.delete(1.0, tk.END)

def open\_file(self):

file\_path = filedialog.askopenfilename(defaultextension=".txt",

filetypes=[("Text files", "\*.txt"), ("All files", "\*.\*")])

if file\_path:

with open(file\_path, "r") as file:

self.text\_area.delete(1.0, tk.END)

self.text\_area.insert(1.0, file.read())

self.root.title(f"Simple Text Editor - {file\_path}")

def save\_file(self):

file\_path = filedialog.asksaveasfilename(defaultextension=".txt",

filetypes=[("Text files", "\*.txt"), ("All files", "\*.\*")])

if file\_path:

with open(file\_path, "w") as file:

file.write(self.text\_area.get(1.0, tk.END))

self.root.title(f"Simple Text Editor - {file\_path}")

def save\_as\_file(self):

self.save\_file()

def exit\_editor(self):

if messagebox.askokcancel("Quit", "Do you really want to quit?"):

self.root.destroy()

# Run the application

if \_\_name\_\_ == "\_\_main\_\_":

root = tk.Tk()

app = TextEditor(root)

root.mainloop()