Advanced Calculator with GUI This project is a modern, feature-rich calculator application built using Python's "tkinter" library for the graphical user interface (GUI). It provides a clean and user-friendly layout, supporting basic and advanced mathematical operations

. Key Features:

1. Basic Arithmetic: Addition, subtraction, multiplication, and division.

2. Advanced Functions: - Square root (√). - Percentage (%). - Backspace (⌫) to delete the last character. 3. Error Handling: Displays clear messages for invalid inputs.

4. Modern Design: Color-coded buttons, responsive layout, and an intuitive interface.

This project serves as an excellent example of GUI development in Python, demonstrating dynamic widget creation, event handling, and application logic. It is suitable for beginners exploring GUI programming or as a foundation for more complex projects.



Code:

import tkinter as tk  
from math import sqrt  
  
# Function to handle button clicks  
def click(event):  
 global expression  
 text = event.widget.cget("text")  
  
 if text == "=":  
 try:  
 # Evaluate the expression  
 result = eval(expression)  
 screen\_var.set(result)  
 expression = str(result)  
 except Exception:  
 screen\_var.set("Error")  
 expression = ""  
 elif text == "C":  
 expression = ""  
 screen\_var.set("")  
 elif text == "⌫":  
 # Backspace functionality  
 expression = expression[:-**1**]  
 screen\_var.set(expression)  
 elif text == "√":  
 try:  
 result = sqrt(float(expression)) if expression else **0** screen\_var.set(result)  
 expression = str(result)  
 except Exception:  
 screen\_var.set("Error")  
 expression = ""  
 else:  
 expression += text  
 screen\_var.set(expression)  
  
# Initialize the application  
root = tk.Tk()  
root.title("Advanced Calculator")  
root.geometry("400x600")  
root.resizable(False**,** False)  
  
expression = ""  
screen\_var = tk.StringVar()  
  
# Display Screen  
screen = tk.Entry(  
 root**,** textvar=screen\_var**,** font=("Arial"**, 24**)**,** bd=**8,** relief=tk.SUNKEN**,** justify=tk.RIGHT**,** bg="#f4f4f4"**,** fg="#333"  
)  
screen.pack(fill=tk.BOTH**,** ipadx=**8,** ipady=**20,** pady=**10**)  
  
# Button layout  
buttons = [  
 ["C"**,** "⌫"**,** "%"**,** "/"]**,** ["7"**,** "8"**,** "9"**,** "\*"]**,** ["4"**,** "5"**,** "6"**,** "-"]**,** ["1"**,** "2"**,** "3"**,** "+"]**,** ["√"**,** "0"**,** "."**,** "="]**,**]  
  
# Button styles  
button\_colors = {  
 "operators": "#ff8c00"**,** "numbers": "#4CAF50"**,** "special": "#f44336"**,** "equal": "#3f51b5"  
}  
  
# Create buttons dynamically  
for row in buttons:  
 button\_frame = tk.Frame(root**,** bg="#ffffff")  
 button\_frame.pack(expand=True**,** fill="both")  
 for char in row:  
 color = (  
 button\_colors["operators"]  
 if char in {"+"**,** "-"**,** "\*"**,** "/"**,** "%"}  
 else button\_colors["special"]  
 if char in {"C"**,** "⌫"**,** "√"}  
 else button\_colors["equal"]  
 if char == "="  
 else button\_colors["numbers"]  
 )  
 button = tk.Button(  
 button\_frame**,** text=char**,** font=("Arial"**, 18**)**,** relief=tk.RAISED**,** border=**0,** bg=color**,** fg="white"**,** activebackground="#ffffff"**,** activeforeground=color**,** )  
 button.pack(side="left"**,** expand=True**,** fill="both"**,** padx=**5,** pady=**5**)  
 button.bind("<Button-1>"**,** click)  
  
# Run the application  
root.mainloop()