**Simple Calculator**

**Introduction**

**Title:** JKH Calculator

The application builds its basic graphical user interface through Tkinter library to allow users access to arithmetic functionality using an easy-to-use interface. The calculator system provides a complete set of operations which includes addition together with subtraction and multiplication together with division and exponentiation and square root and modulus operations.

**User Flow & Interface Design**

**Application Launch**

**Description:**

Users can initiate the calculator program after opening the application.

When the application launches JKH Calculator opens in a dialog window format.

**The interface consists of**:

The user interface contains two numerical text boxes.

Users can select arithmetic operations by using a dropdown system found in the application.

The "Calculate" button enables operation execution within the application.

A simple click on the "Reset" button empties every text box that the application contains.

The application shows the results through a displayed label.

**Visual Elements for JKH Calculator**

**Window Title:**

**Title:** "JKH Calculator" at the top of the window.

**Input Fields:**

The window features two Number1 and Number2 input text boxes that display default value "0".

**Drop-down Menu:**

The drop-down menu of the application shows arithmetic operation choices (such as Add, Subtract, Multiply).

**Buttons:**

Execution of the operation occurs when users click the "Calculate" button present in the interface.

The application includes a "Reset" button to let users restore default calculations while removing their current input data.

**Result Display:**

The display screen either displays the computational outcome as Result: 10 or Error!.

**Layout:**

The interface design follows a centered format to present all elements correctly through uncluttered spaces.  
**User Input**

**Description:**

Users need to input numbers through the text boxes which the application software provides for access.

The arithmetic operation which users should select comes from a drop-down menu that includes "Add," "Subtract," "Multiply" and additional options.  
User input depends only on the square root calculation since it requires first field data entry.

**Visual Elements:**

User typing numbers into input fields.

Drop-down menu showing available operations.  
**Calculation Process**

**Description:**

* The user triggers the "Calculate" button activation.
* All the program obtains data from the stated input boxes as part of its operation.
* It checks the selected operation.
* The program performs the calculation.
* The Result label displays the calculated results on the interface.
* In case there are issues with division by zero the application shows clear error notifications.

**Visual Elements:**

* The program uses button when the user clicks the "Calculate" button.

The Result label displays calculations either as outcomes or error indications.  
**Displaying the Result**

**Description:**

The user operation of the "Calculate" button causes the system to display the result within the "Result" label.

Number1 serves as the only value during calculation if Number1 contains the square root computation.

Incorrect user input containing non-numbers or any attempt to perform zero division results in an error message rather than a result display.

**Visual Elements:**

When users click "Calculate" the "Result" field presents the operation computation which displays "Result: 15" during addition operations.

Users will see an error message displayed in the label when any errors occur during operation. Division by zero".  
**User Actions:** Enter numbers → Select operation → Click Calculate → View result → Reset if needed.

The features include minimal arithmetics along with automatic error management and the ability to reset the system.

**Outcome:** Simple, user-friendly calculator.  
**simple calculator code**import tkinter as tk

import math

# Function to handle the operation

def calculate():

    try:

        num1 = float(entry1.get())

        num2 = float(entry2.get())

        operation = operation\_var.get()

        if operation == "Add":

            result = num1 + num2

        elif operation == "Subtract":

            result = num1 - num2

        elif operation == "Multiply":

            result = num1 \* num2

        elif operation == "Divide":

            if num2 == 0:

                result = "Error! Division by zero."

            else:

                result = num1 / num2

        elif operation == "Exponentiate":

            result = num1 \*\* num2

        elif operation == "Square Root":

            result = math.sqrt(num1)

        elif operation == "Modulus":

            result = num1 % num2

        # Display result

        result\_label.config(text="Result: " + str(result))

    except ValueError:

        result\_label.config(text="Error! Invalid input.")

# Function to clear the entries and result

def reset():

    entry1.delete(0, tk.END)

    entry2.delete(0, tk.END)

    result\_label.config(text="Result: ")

# Create the main window

root = tk.Tk()

root.title("JKH Calculator")

# Create and place labels and entry fields

tk.Label(root, text="Number1:").grid(row=0, column=0)

entry1 = tk.Entry(root)

entry1.grid(row=0, column=1)

tk.Label(root, text="Number2:").grid(row=1, column=0)

entry2 = tk.Entry(root)

entry2.grid(row=1, column=1)

# Create operation choices

operation\_var = tk.StringVar(root)

operation\_var.set("Add")

operations\_menu = tk.OptionMenu(root, operation\_var, "Add", "Subtract", "Multiply", "Divide", "Exponentiate", "Square Root", "Modulus")

operations\_menu.grid(row=2, column=0, columnspan=2)

# Create calculate and reset buttons

calculate\_button = tk.Button(root, text="Calculate", command=calculate)

calculate\_button.grid(row=3, column=0, pady=10)

reset\_button = tk.Button(root, text="Reset", command=reset)

reset\_button.grid(row=3, column=1, pady=10)

# Create result label

result\_label = tk.Label(root, text="Result: ")

result\_label.grid(row=4, column=0, columnspan=2)

# Run the GUI event loop

root.mainloop()  
**calculator working image**

   
