# ভয়েস ক্যালকুলেটর কোড ব্যাখ্যা (বাংলা) - Step by Step ছোট ছোট ধাপে

## ১. লাইব্রেরি Import করা

import customtkinter as ctk  
import re  
import speech\_recognition as sr  
import threading  
import time  
import math

* **customtkinter (ctk):** Tkinter-এর modern version। Dark/Light mode, সুন্দর button ও entry তৈরি করতে সহায়ক।
* **re:** Regular expressions; input validation এবং special character handle করার জন্য।
* **speech\_recognition (sr):** ভয়েস ইনপুট নেয়ার জন্য।
* **threading:** UI hang না করে background এ voice recognition চালানোর জন্য।
* **time:** Animation delay ও live typing effect এর জন্য।
* **math:** √ (square root) এবং অন্যান্য math functions handle করতে।

যদি কোন লাইব্রেরি import না করা হয়, কোড run হবে না। প্রতিটি লাইব্রেরি স্পেসিফিক কাজের জন্য দরকার।

## ২. Safe Evaluation ফাংশন (safe\_eval)

def safe\_eval(expr):  
 try:  
 expr = expr.replace("√", "math.sqrt")  
 expr = expr.replace("²", "\*\*2")  
 allowed = re.compile(r'^[0-9\.+\-\*/() mathsqrt]+$')  
 if not allowed.match(expr):  
 return "ERROR"  
 result = eval(expr)  
 if isinstance(result, float):  
 result = round(result, 10)  
 return result  
 except:  
 return "ERROR"

* **কারণ:** সরাসরি eval ব্যবহার করলে unsafe হতে পারে। এখানে শুধু safe characters allow করা হয়েছে।
* **√ → math.sqrt, ² → 2:** Python execute করার জন্য।
* **ERROR handling:** ভুল input হলে user-friendly “ERROR” দেখাবে।

যদি allowed regex ঠিক না রাখা হতো, security risk হতে পারত।

## ৩. Button Click Handler (on\_button\_click)

def on\_button\_click(value):  
 current\_text = entry\_field.get()  
  
 if value == "C":  
 entry\_field.delete(0, ctk.END) # Clear সব input  
  
 elif value == "DEL":  
 if current\_text == "ERROR":  
 entry\_field.delete(0, ctk.END)  
 elif len(current\_text) > 0:  
 entry\_field.delete(len(current\_text)-1, ctk.END) # last character delete  
  
 elif value == "=":  
 if current\_text.strip() == "": return  
 result = safe\_eval(current\_text) # expression calculate  
 entry\_field.delete(0, ctk.END)  
 entry\_field.insert(ctk.END, str(result))  
  
 elif value == "x²":  
 entry\_field.insert(ctk.END, "²") # square operator add  
  
 elif value == "√":  
 entry\_field.insert(ctk.END, "√") # square root operator add  
  
 else:  
 entry\_field.insert(ctk.END, value) # normal number/operator add  
  
 entry\_field.xview\_moveto(1) # Cursor সবসময় শেষে থাকবে

* প্রতিটি button আলাদা কাজ করে।
* Cursor সবসময় শেষে রাখলে user experience ভালো হয়।
* **অপশনাল:** যদি Cursor না শেষের দিকে রাখা হতো, user input confuse হতো।

## ৪. Voice Button Animation

def animate\_voice\_button(stop\_event):  
 colors = ["#1f6aa5", "#2f82c5", "#4da3e0"]  
 i = 0  
 while not stop\_event.is\_set():  
 voice\_btn.configure(fg\_color=colors[i % len(colors)])  
 i += 1  
 time.sleep(0.3)  
 voice\_btn.configure(fg\_color="#2c506b") # Reset

* **কেন:** Listening অবস্থায় user দেখবে button animation।
* **stop\_event:** Thread safe animation বন্ধ করার জন্য।
* **অন্য color দিলে:** aesthetic impact পরিবর্তন হতে পারে।

## ৫. Voice Command Handling

def voice\_command\_thread():  
 stop\_event = threading.Event()  
 animation\_thread = threading.Thread(target=animate\_voice\_button, args=(stop\_event,), daemon=True)  
 animation\_thread.start()  
  
 voice\_btn.configure(text="Listening...", font=("Consolas", 16, "bold"))  
  
 recognizer = sr.Recognizer()  
 try:  
 with sr.Microphone() as source:  
 audio\_data = recognizer.listen(source, timeout=5, phrase\_time\_limit=8)  
 command\_text = recognizer.recognize\_google(audio\_data, language="bn-BD")  
  
 replacements = {"যোগ":"+", "বিয়োগ":"-", "বিয়োগ":"-", "গুণ":"\*", "ভাগ":"/",  
 "plus":"+", "minus":"-", "multiply":"\*", "times":"\*", "divide":"/",  
 "square":"²", "√":"√", "root":"√"}  
  
 cmd = command\_text.lower()  
 for word, symbol in replacements.items():  
 cmd = cmd.replace(word, symbol)  
  
 cmd = re.sub(r"[^0-9\+\-\\*\/\(\)√²\.]", "", cmd)  
  
 for char in cmd:  
 entry\_field.insert(ctk.END, char)  
 entry\_field.update()  
 time.sleep(0.03)  
  
 except sr.UnknownValueError:  
 entry\_field.insert(ctk.END, " [Didn't catch that]")  
 except sr.RequestError:  
 entry\_field.insert(ctk.END, " [Speech error]")  
 except Exception as e:  
 entry\_field.insert(ctk.END, " [Error]")  
  
 stop\_event.set()  
 voice\_btn.configure(text="🎤 Voice", font=("Consolas", 20, "bold"))

* **Step by Step:**
  1. Threading দিয়ে background এ voice recognition।
  2. Listening হলে button text change।
  3. Google API দিয়ে speech-to-text।
  4. Replace Bengali/English words to operators।
  5. Valid character filter।
  6. Entry field update digit-by-digit।
  7. Error handling।

যদি animation বা threading না থাকতো, UI hang হতো।

## ৬. GUI Setup

ctk.set\_appearance\_mode("dark")  
ctk.set\_default\_color\_theme("blue")  
root = ctk.CTk()  
root.title("Voice Calculator")

* Dark mode, theme setup।
* User-friendly appearance।

### Entry Field

entry\_field = ctk.CTkEntry(root, width=280, height=50, font=("Consolas", 28, "bold"), justify="right", fg\_color="#BBB", text\_color="#000")  
entry\_field.grid(row=0, column=0, columnspan=4, pady=10, padx=5)

* Right aligned, readable font।
* Background color & height/width user-friendly।

### Buttons

* Top row: C, Voice, DEL
* Numbers & operators grid layout
* Lambda binding: lambda t=text: on\_button\_click(t) ensures correct function call.

## ৭. ভবিষ্যৎ উন্নতি

* বাংলা numbers recognition
* Calculation history দেখানো
* Keyboard input support
* Advanced math functions (trig, log, factorial)
* User-friendly error messages

প্রতিটি step ছোট ছোট ভাগে করলে debugging সহজ হয়।