

```
def show_message_box(self, title, message):
    """কাস্টম মেসেজ বক্স প্রদর্শন করে (alert() এর পরিবর্তে)"""
    msg_box = tk.Toplevel(self.root)
    msg_box.title(title)
    msg_box.geometry("250x150")
    msg_box.transient(self.root) # মূল উইন্ডোর উপরে রাখুন
    msg_box.grab_set() # মূল উইন্ডোতে ক্লিক করা যাবে না

    tk.Label(msg_box, text=message, font=self.text_font, wraplength=200).pack(pady=20)
    tk.Button(msg_box, text="OK", command=msg_box.destroy, font=self.text_font, bg="#3498db", fg=
        activebackground="#2980b9").pack(pady=10)

    self.root.wait_window(msg_box) # মেসেজ বক্স বন্ধ না হওয়া পর্যন্ত অপেক্ষা করুন

if __name__ == "__main__":
    root = tk.Tk()
    app = ConverterApp(root)
    root.mainloop()
```

gimini x

```
Users\infom\convartertools\pythonProject1\venv\Scripts\python.exe C:\Users\infom\convartertools\pyth
```

```
# "কিভাবে রূপান্তর করা হলো" ব্যঙ্গ
explanation_label = tk.Label(result_box_frame, text="কিভাবে রূপান্তর করা হলো:", font=self.text_font,
    fg="#2c3e50", bg="#bdc3c7", anchor="n")
explanation_label.pack(fill="x", pady=5)

# ব্যাখ্যা প্রদর্শনের জন্য টেক্সট উইজেট
explanation_text_widget = tk.Text(result_box_frame, font=self.text_font, fg="#34495e", bg="#ecf0f1",
    wrap="word", height=10, relief="solid", bd=1, padx=5, pady=5)
explanation_text_widget.pack(fill="both", expand=True, pady=5)
explanation_text_widget.insert(tk.END, explanation)
explanation_text_widget.config(state="disabled") # শুধুমাত্র পড়ার জন্য সেট করুন

# "Back Home" বটাম
back_button = tk.Button(results_frame, text="Back Home 🏠", font=self.button_font, command=self.show_home_page,
    bg="#3498db", fg="white", activebackground="#2980b9", relief="raised", bd=2)
back_button.pack(pady=10) # pady কমানো হয়েছে
```

usage

```
def show_message_box(self, title, message):
    """কাস্টম মেসেজ বক্স প্রদর্শন করে (alert() এর পরিবর্তে)"""
    msg_box = tk.Toplevel(self.root)
    msg_box.title(title)
    msg_box.geometry("250x150")
    msg_box.transient(self.root) # মূল উইন্ডোর উপরে রাখুন
    msg_box.grab_set() # মূল উইন্ডোতে ক্লিক করা যাবে না

    tk.Label(msg_box, text=message, font=self.text_font, wraplength=200).pack(pady=20)
    tk.Button(msg_box, text="OK", command=msg_box.destroy, font=self.text_font, bg="#3498db", fg="white",
        activebackground="#2980b9").pack(pady=10)

    self.root.wait_window(msg_box) # মেসেজ বক্স বন্ধ না হওয়া পর্যন্ত অপেক্ষা করুন
```

```
# "কিভাবে রূপান্তর করা হলো" ব্যঙ্গ
explanation_label = tk.Label(result_box_frame, text="কিভাবে রূপান্তর করা হলো:", font=self.text_font,
    fg="#2c3e50", bg="#bdc3c7", anchor="n")
explanation_label.pack(fill="x", pady=5)

# ব্যাখ্যা প্রদর্শনের জন্য টেক্সট উইজেট
explanation_text_widget = tk.Text(result_box_frame, font=self.text_font, fg="#34495e", bg="#ecf0f1",
    wrap="word", height=10, relief="solid", bd=1, padx=5, pady=5)
explanation_text_widget.pack(fill="both", expand=True, pady=5)
explanation_text_widget.insert(tk.END, explanation)
explanation_text_widget.config(state="disabled") # শুধুমাত্র পড়ার জন্য সেট করুন

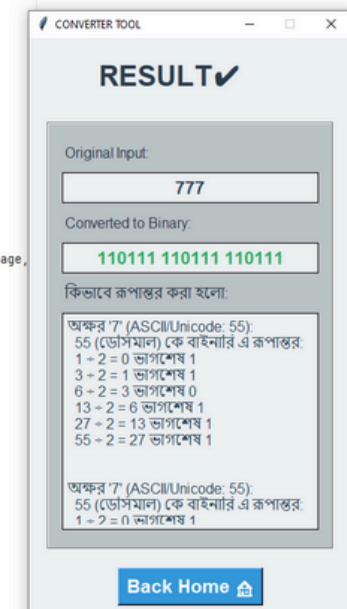
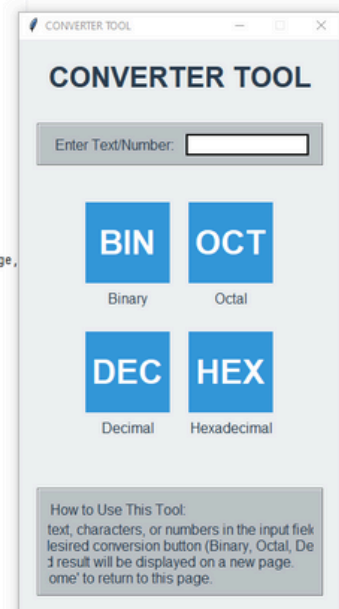
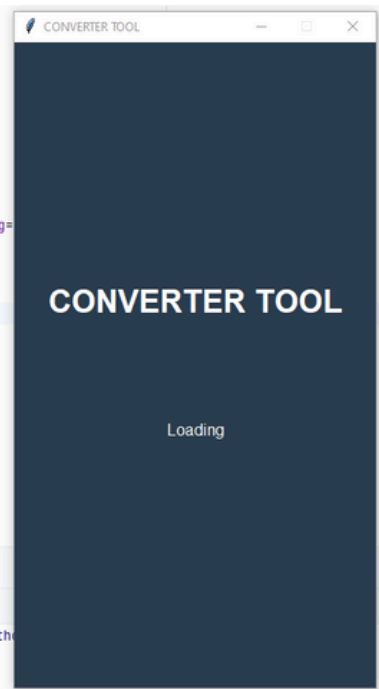
# "Back Home" বটাম
back_button = tk.Button(results_frame, text="Back Home 🏠", font=self.button_font, command=self.show_home_page,
    bg="#3498db", fg="white", activebackground="#2980b9", relief="raised", bd=2)
back_button.pack(pady=10) # pady কমানো হয়েছে
```

usage

```
def show_message_box(self, title, message):
    """কাস্টম মেসেজ বক্স প্রদর্শন করে (alert() এর পরিবর্তে)"""
    msg_box = tk.Toplevel(self.root)
    msg_box.title(title)
    msg_box.geometry("250x150")
    msg_box.transient(self.root) # মূল উইন্ডোর উপরে রাখুন
    msg_box.grab_set() # মূল উইন্ডোতে ক্লিক করা যাবে না

    tk.Label(msg_box, text=message, font=self.text_font, wraplength=200).pack(pady=20)
    tk.Button(msg_box, text="OK", command=msg_box.destroy, font=self.text_font, bg="#3498db", fg="white",
        activebackground="#2980b9").pack(pady=10)

    self.root.wait_window(msg_box) # মেসেজ বক্স বন্ধ না হওয়া পর্যন্ত অপেক্ষা করুন
```



```

import tkinter as tk
from tkinter import ttk
import tkinter.font as tkFont
import time
1 usage
class ConverterApp:
    def __init__(self, root):
        self.root = root
        self.root.title("CONVERTER TOOL")

        # অ্যান্ড্রয়েড মোবাইল স্ক্রিন সাইজ (উদাহরণস্বরূপ 360x640)
        self.screen_width = 360
        self.screen_height = 640
        self.root.geometry(f"{self.screen_width}x{self.screen_height}")
        self.root.resizable(False, False) # উইন্ডো রিসাইজ করা যাবে না

        # ফন্ট সেটআপ
        self.title_font = tkFont.Font(family="Arial", size=24, weight="bold")
        self.button_font = tkFont.Font(family="Arial", size=16, weight="bold")
        self.text_font = tkFont.Font(family="Arial", size=12)
        self.result_font = tkFont.Font(family="Arial", size=14, weight="bold")

        # বর্তমান ফ্রেম ট্র্যাক করার জন্য
        self.current_frame = None

        self.show_splash_screen()

3 usages
    def clear_frame(self):
        """বর্তমান ফ্রেমের সমস্ত উইজেট পরিষ্কার করে।"""
        if self.current_frame:
            for widget in self.current_frame.winfo_children():
                widget.destroy()
            self.current_frame.destroy()
        self.current_frame = None

    def show_splash_screen(self):
        """স্প্লাশ স্ক্রিন প্রদর্শন করে।"""
        self.clear_frame()
        splash_frame = tk.Frame(self.root, bg="#2c3e50") # গাঢ় নীল/ধূসর ব্যাকগ্রাউন্ড
        splash_frame.pack(fill="both", expand=True)
        self.current_frame = splash_frame

        # "CONVERTER TOOL" লেখা
        title_label = tk.Label(splash_frame, text="CONVERTER TOOL", font=self.title_font, fg="white", bg="#2c3e50")
        title_label.place(relx=0.5, rely=0.4, anchor="center")

        # লোডিং টেক্সট
        self.loading_label = tk.Label(splash_frame, text="Loading...", font=self.text_font, fg="white", bg="#2c3e50")
        self.loading_label.place(relx=0.5, rely=0.6, anchor="center")

        # লোডিং অ্যানিমেশন (৫ সেকেন্ড)
        self.loading_dots = 0
        self.update_loading_text()
        self.root.after(5000, self.show_home_page) # ৫ সেকেন্ড পর হোম পেজে যাবে

2 usages
    def update_loading_text(self):
        """লোডিং টেক্সট অ্যানিমেশন আপডেট করে।"""
        dots = "." * (self.loading_dots % 4)
        self.loading_label.config(text=f"Loading{dots}")
        self.loading_dots += 1
        if self.current_frame == self.root.winfo_children()[0]: # নিশ্চিত করুন যে এটি এখনও স্প্লাশ স্ক্রিন
            self.root.after(500, self.update_loading_text) # প্রতি 0.5 সেকেন্ডে আপডেট

2 usages
    def show_home_page(self):
        """হোম পেজ প্রদর্শন করে।"""
        self.clear_frame()
        home_frame = tk.Frame(self.root, bg="#ecf0f1") # হালকা ধূসর ব্যাকগ্রাউন্ড
        home_frame.pack(fill="both", expand=True)
        self.current_frame = home_frame

```

```

def show_home_page(self):
    """হোম পেজ প্রদর্শন করে।"""
    self.clear_frame()
    home_frame = tk.Frame(self.root, bg="#ecf0f1") # হালকা নীল ব্যাকগ্রাউন্ড
    home_frame.pack(fill="both", expand=True)
    self.current_frame = home_frame

    # "CONVERTER TOOL" শিরোনাম
    title_label = tk.Label(home_frame, text="CONVERTER TOOL", font=self.title_font, fg="#2c3e50", bg="#ecf0f1")
    title_label.pack(pady=20)

    # ইনপুট ফিল্ড
    input_frame = tk.Frame(home_frame, bg="#bdc3c7", padx=10, pady=10, relief="groove",
                           bd=2) # ইনপুট ফ্রেমের জন্য আলগা ব্যাকগ্রাউন্ড
    input_frame.pack(pady=10, padx=20, fill="x")

    input_label = tk.Label(input_frame, text="Enter Text/Number:", font=self.text_font, fg="#2c3e50", bg="#bdc3c7")
    input_label.pack(side="left", padx=5)

    self.input_entry = tk.Entry(input_frame, font=self.text_font, width=25, bd=2, relief="solid")
    self.input_entry.pack(side="right", expand=True, fill="x", padx=5)

    # বাটন ফ্রেম
    button_frame = tk.Frame(home_frame, bg="#ecf0f1")
    button_frame.pack(pady=20)

    # বাটন স্টাইল
    button_bg = "#3498db" # নীল
    button_fg = "white"
    button_active_bg = "#2980b9" # গাঢ় নীল

    # বাইনারি বাটন
    self.create_conversion_button(button_frame, icon_text="BIN", button_name="Binary", row=0, col=0, button_bg, button_fg,
                                  self.convert_to_binary)

    # অক্টাল বাটন
    self.create_conversion_button(button_frame, icon_text="OCT", button_name="Octal", row=0, col=1, button_bg, button_fg,
                                  self.convert_to_octal)

    self.create_conversion_button(button_frame, icon_text="DEC", button_name="Decimal", row=1, col=0, button_bg, button_fg,
                                  self.convert_to_decimal)

    # হেক্সাডেসিমাল বাটন
    self.create_conversion_button(button_frame, icon_text="HEX", button_name="Hexadecimal", row=1, col=1, button_bg, button_fg,
                                  self.convert_to_hexadecimal)

    # "How to work" ব্যাকগ্রাউন্ড
    how_to_work_frame = tk.Frame(home_frame, bg="#bdc3c7", padx=10, pady=10, relief="groove", bd=2)
    how_to_work_frame.pack(pady=20, padx=20, fill="x")

    how_to_work_label = tk.Label(how_to_work_frame, text="How to Use This Tool:", font=self.text_font, fg="#2c3e50",
                                 bg="#bdc3c7", anchor="w")
    how_to_work_label.pack(fill="x")

    instructions_text = """
1. Enter any English text, characters, or numbers in the input field.
2. Click on the desired conversion button (Binary, Octal, Decimal, Hexadecimal).
3. The converted result will be displayed on a new page.
4. Click 'Back Home' to return to this page.
"""

    instructions_label = tk.Label(how_to_work_frame, text=instructions_text.strip(), font=self.text_font,
                                 fg="#2c3e50", bg="#bdc3c7", justify="left")
    instructions_label.pack(fill="x")

    # এক্সিট বাটন
    exit_button = tk.Button(home_frame, text="Exit X", font=self.text_font, command=self.root.destroy, bg="#e74c3c",
                           fg="white", activebackground="#c0392b", relief="raised", bd=2)
    exit_button.pack(pady=10)

usages
def create_conversion_button(self, parent_frame, icon_text, button_name, row, col, bg_color, fg_color,
                             active_bg_color, command_func):
    """হোম পেজে একটি কনভার্সন বাটন তৈরি করে।"""
    btn_frame = tk.Frame(parent_frame, bg=parent_frame['bg'])
    btn_frame.grid(row=row, column=col, padx=10, pady=10)

1 usage
def convert_to_decimal(self):
    self.perform_conversion("decimal")

1 usage
def convert_to_hexadecimal(self):
    self.perform_conversion("hexadecimal")

```

6 usages

```

converted_result = " ".join(converted_result_parts)
explanation = "\n".join(explanation_parts).strip() # অতিরিক্ত নতুন লাইন সরান

self.show_results_page(input_data, conversion_type.capitalize(), converted_result, explanation)

1 usage
def show_results_page(self, original_input, conversion_type, result, explanation):
    """ফলাফল পেজ প্রদর্শন করে।"""
    self.clear_frame()
    results_frame = tk.Frame(self.root, bg="#ecf0f1") # ফলকা ধূসর ব্যাকগ্রাউন্ড
    results_frame.pack(fill="both", expand=True)
    self.current_frame = results_frame

    # "CONVERTER TOOL" শিরোনাম
    title_label = tk.Label(results_frame, text="RESULT✔", font=self.title_font, fg="#2c3e50", bg="#ecf0f1")
    title_label.pack(pady=20)

    # ফলাফল ব্যাক
    result_box_frame = tk.Frame(results_frame, bg="#bdc3c7", padx=15, pady=15, relief="groove", bd=2)
    result_box_frame.pack(pady=10, padx=20, fill="both", expand=True) # pady কমানো হয়েছে

    original_label = tk.Label(result_box_frame, text="Original Input:", font=self.text_font, fg="#2c3e50",
                              bg="#bdc3c7", anchor="w")
    original_label.pack(fill="x", pady=5)
    original_value_label = tk.Label(result_box_frame, text=original_input, font=self.result_font, fg="#34495e",
                                    bg="#ecf0f1", wraplength=self.screen_width - 80, justify="left", relief="solid",
                                    bd=1, padx=5, pady=5)
    original_value_label.pack(fill="x", pady=5)

    type_label = tk.Label(result_box_frame, text=f"Converted to {conversion_type}:", font=self.text_font,
                          fg="#2c3e50", bg="#bdc3c7", anchor="w")
    type_label.pack(fill="x", pady=5)
    result_value_label = tk.Label(result_box_frame, text=result, font=self.result_font, fg="#27ae60", bg="#ecf0f1",
                                  wraplength=self.screen_width - 80, justify="left", relief="solid", bd=1, padx=5,
                                  pady=5)
    result_value_label.pack(fill="x", pady=5)

    # "কিভাবে রূপান্তর করা হলো" ব্যাক
    explanation_label = tk.Label(result_box_frame, text="কিভাবে রূপান্তর করা হলো:", font=self.text_font,
                                  fg="#2c3e50", bg="#bdc3c7", anchor="w")
    explanation_label.pack(fill="x", pady=5)

    # ব্যাখ্যা প্রদর্শনের জন্য টেক্সট উইজেট
    explanation_text_widget = tk.Text(result_box_frame, font=self.text_font, fg="#34495e", bg="#ecf0f1",
                                       wrap="word", height=10, relief="solid", bd=1, padx=5, pady=5)
    explanation_text_widget.pack(fill="both", expand=True, pady=5)
    explanation_text_widget.insert(tk.END, explanation)
    explanation_text_widget.config(state="disabled") # শুধুমাত্র পড়ার জন্য সেট করুন

    # "Back Home" বাটন
    back_button = tk.Button(results_frame, text="Back Home 🏠", font=self.button_font, command=self.show_home_page,
                             bg="#3498db", fg="white", activebackground="#2980b9", relief="raised", bd=2)
    back_button.pack(pady=10) # pady কমানো হয়েছে

1 usage
def show_message_box(self, title, message):
    """কাস্টম মেসেজ বক্স প্রদর্শন করে (alert() এর পরিবর্তে)"""
    msg_box = tk.Toplevel(self.root)
    msg_box.title(title)
    msg_box.geometry("250x150")
    msg_box.transient(self.root) # মূল উইন্ডোর উপরে রাখুন
    msg_box.grab_set() # মূল উইন্ডোতে ক্লিক করা যাবে না

    tk.Label(msg_box, text=message, font=self.text_font, wraplength=200).pack(pady=20)
    tk.Button(msg_box, text="OK", command=msg_box.destroy, font=self.text_font, bg="#3498db", fg="white",
              activebackground="#2980b9").pack(pady=10)

    self.root.wait_window(msg_box) # মেসেজ বক্স বন্ধ না হওয়া পর্যন্ত অপেক্ষা করুন

if __name__ == "__main__":
    root = tk.Tk()
    app = ConverterApp(root)
    root.mainloop()

```