GUI PROJECT IN ITE121 Intermediate Programming

SUBMITTED BY: EXZUR G. LAMANILAO

BSCS-1A

SUBMITTED TO: MRS. UNIFE O. CAGAS

DESCRIPTION OF THE PROJECT

The Universal Unit Converter is a versatile graphical user interface (GUI) application designed to streamline and simplify unit conversions across various measurement systems. Whether you're a student, a professional, or someone who frequently deals with measurements in daily tasks, this intuitive tool offers an efficient solution to convert units seamlessly.

WHAT THE PROJECT IS ALL ABOUT?

The project is a Graphical User Interface (GUI) application focused on unit conversion. It provides a user-friendly platform for individuals to easily convert measurements from one unit to another across various categories such as length, weight, volume, temperature. The core functionality revolves around inputting a measurement in a certain unit and instantly getting the equivalent value in another unit of choice, all within the application's interface.

The project aims to simplify the process of unit conversion for users across different backgrounds and professions. It caters to students, professionals, and general users who frequently encounter the need to convert units in their academic, professional, or personal endeavors. Whether it's converting units for assignments, projects, recipes, construction plans, or travel itineraries, the GUI application offers a convenient solution.

BENEFIT

The project simplifies unit conversions, saving time and ensuring accuracy for users across different fields and activities.

IMPORTANCE

This project is important as it addresses a fundamental need in various domains where precise unit conversions are required. By providing a user-friendly interface for quick and accurate conversions, it streamlines workflows, reduces errors, and enhances productivity. Whether used by students tackling math problems, professionals in engineering or science fields, or individuals in everyday tasks like cooking or DIY projects, this tool simplifies the often cumbersome process of converting between different units, ultimately contributing to smoother operations and better outcomes in a wide range of endeavors.

PURPOSE

The purpose of this project is to develop a graphical user interface (GUI) application that simplifies the process of unit conversion across various measurement systems. By providing an intuitive platform for users to input measurements and instantly obtain converted values, the project aims to streamline tasks that require converting between different units, such as length, weight, volume, temperature. This tool serves to enhance efficiency, accuracy, and convenience for individuals across different fields and activities, ultimately enabling smoother workflows and improved outcomes in academic, professional, and personal endeavors.

FEATURES

1. GUI built using Tkinter library.

2. Customized GUI elements using a custom tkinter module (customtkinter).

3. Unit Converter application for Length, Mass, Volume, and Temperature conversions.

4. Supported units for each category:

   - Length: Meter, Centimeter, Foot

   - Mass: Kilogram, Gram, Pound

   - Volume: Liter, Milliliter, Gallon

   - Temperature: Celsius, Fahrenheit, Kelvin

5. Input fields for entering the value to be converted.

6. Dropdown menus for selecting the type of conversion (Length, Mass, Volume, Temperature) and the units to convert from and to.

7. Conversion functionality implemented for each unit type.

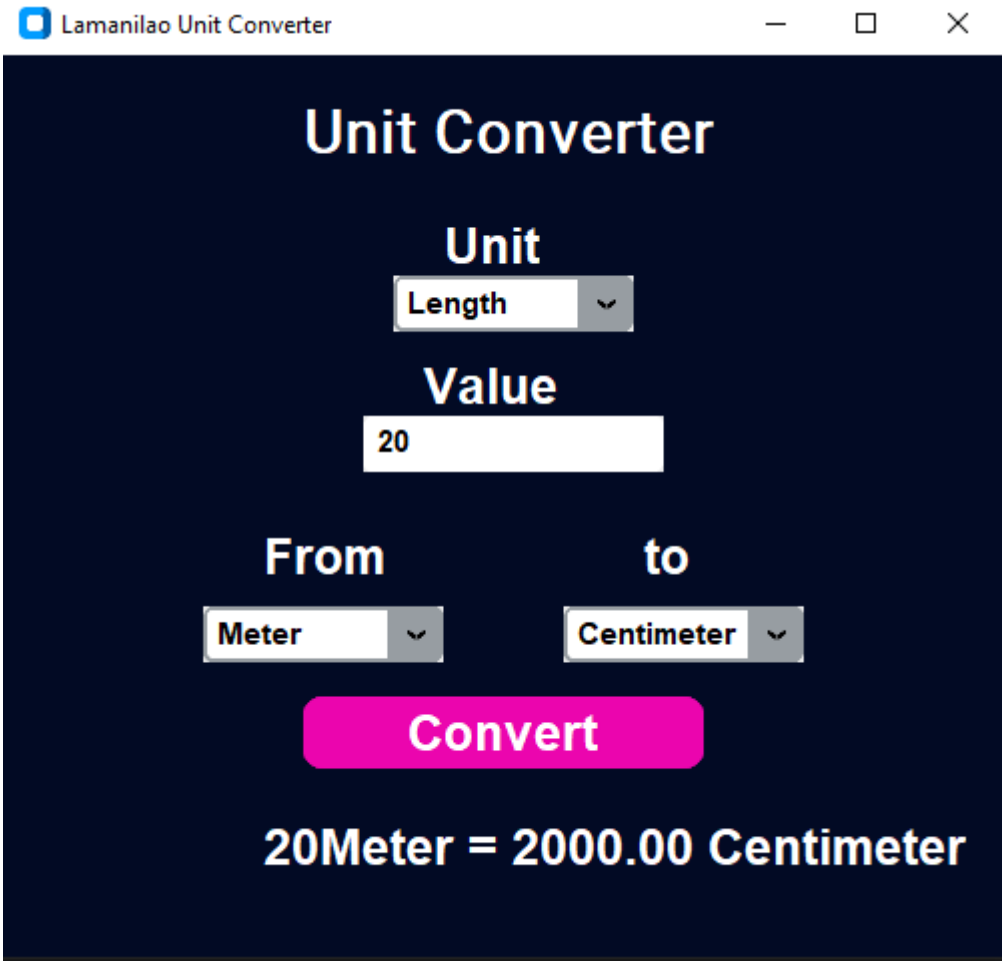
8. Error handling for invalid input values.

9. Result display area to show the converted value.

10. Responsive layout with proper alignment and spacing.

11. Stylish design with customizable colors and fonts.

FUNCTIONALITIES



This dropdown menu of unit, serves as the options of the list of units the user wants to use.

This part is where the user put the value they want to be converted

This is another dropdown menu in which the user can pick what unit of measurement the values is from and what unit they are going to be converted to.

The last part is the Convert button and the result label. The convert button act as the switch to convert the certain value that is given and is shown on the bottom part of the screen.

CODE

```
from tkinter import *
from tkinter import messagebox

import customtkinter as ctk

app = ctk.CTk()
app.title('Lamanilao Unit Converter')
app.geometry('500x450')
app.config(bg='#020a24')

font1 = ('Roboto', 30, 'bold')
font2 = ('Arial', 25, 'bold')
font3 = ('Arial', 15, 'bold')

unit_options = ['Length', 'Mass', 'Volume',
'Temperature']  # Added 'Temperature' to unit options
length_options = ['Meter', 'Centimeter', 'Foot']
mass_options = ['Kilogram', 'Gram', 'Pound']
volume_options = ['Liter', 'Milliliter', 'Gallon']
temperature_options = ['Celsius', 'Fahrenheit', 'Kelvin']  #
Added temperature options
variable1 = ctk.StringVar()
variable2 = ctk.StringVar()
variable3 = ctk.StringVar()

def convert():
    length_factors = {'Meter': 1, 'Centimeter': 0.01,
'Foot': 0.3048}
    mass_factors = {'Kilogram': 1, 'Gram': 0.0001, 'Pound':
0.453592}
    volume_factors = {'Liter': 1, 'Milliliter': 0.001,
'Gallon': 3.78541}
    temperature_factors = {'Celsius': (lambda x: x),
'Fahrenheit': (lambda x: (x - 32) * 5 / 9), 'Kelvin':
(lambda x: x - 273.15)}
```

```
    try:
        if variable1.get() == 'Length':
            meters = float(value_entry.get()) *
length_factors[variable2.get()]
            converted_value = meters /
length_factors[variable3.get()]
        elif variable1.get() == 'Mass':
            kilograms = float(value_entry.get()) *
mass_factors[variable2.get()]
            converted_value = kilograms /
mass_factors[variable3.get()]
        elif variable1.get() == 'Volume':
            liters = float(value_entry.get()) *
volume_factors[variable2.get()]
            converted_value = liters /
volume_factors[variable3.get()]
        elif variable1.get() == 'Temperature':  # Added
condition for temperature conversion
            original_value = float(value_entry.get())
            converted_value =
temperature_factors[variable3.get()](original_value)
            converted_value =
temperature_factors[variable2.get()](converted_value)
        result_label.configure(text=f'{value_entry.get()}{va
riable2.get()} = {converted_value:.2f} {variable3.get()}')
    except ValueError:
        messagebox.showerror('Error', 'Enter valid values!')

title_label = ctk.CTkLabel(app, font=font1, text='Unit
Converter', text_color='#fff', bg_color='#020a24')
title_label.place(x=150, y=20)

unit_label = ctk.CTkLabel(app, font=font2, text='Unit',
text_color='#fff', bg_color='#020a24')
unit_label.place(x=220, y=80)

unit_option = ctk.CTkComboBox(app, font=font3,
text_color='#000', fg_color='#fff',
```

```
dropdown_hover_color='#06911f', values=unit_options,
variable=variable1, width=120)
unit_option.place(x=195, y=110)

from_label = ctk.CTkLabel(app, font=font2, text='From',
text_color='#fff', bg_color='#020a24')
from_label.place(x=130, y=235)

from_option = ctk.CTkComboBox(app, font=font3,
text_color='#000', fg_color='#fff',
dropdown_hover_color='#06911f', variable=variable2,
width=120)
from_option.place(x=100, y=275)

to_label = ctk.CTkLabel(app, font=font2, text='to',
text_color='#fff', bg_color='#020a24')
to_label.place(x=320, y=235)

to_option = ctk.CTkComboBox(app, font=font3,
text_color='#000', fg_color='#fff',
dropdown_hover_color='#06911f', variable=variable3,
width=120)
to_option.place(x=280, y=275)

value_label = ctk.CTkLabel(app, font=font2, text='Value',
text_color='#fff', bg_color='#020a24')
value_label.place(x=210, y=150)

value_entry = ctk.CTkEntry(app, font=font3,
text_color='#000', fg_color='#fff', border_color='#fff',
width=150)
value_entry.place(x=180, y=180)

convert_button = ctk.CTkButton(app, command=convert,
font=font2, text_color='#fff', text='Convert',
fg_color='#eb05ae', hover_color='#a8057d',
bg_color='#020a24', cursor='hand2', corner_radius=10,
width=200)
```

```
convert_button.place(x=150, y=320)

result_label = ctk.CTkLabel(app, font=font2, text=' ',
text_color='#fff', bg_color='#020a24')
result_label.place(x=130, y=380)

def update_options(*args):
    if variable1.get() == 'Length':
        from_option.configure(values=length_options)
        to_option.configure(values=length_options)
        from_option.set('Meter')
        to_option.set('Centimeter')
    elif variable1.get() == 'Mass':
        from_option.configure(values=mass_options)
        to_option.configure(values=mass_options)
        from_option.set('Kilogram')
        to_option.set('Gram')
    elif variable1.get() == 'Volume':
        from_option.configure(values=volume_options)
        to_option.configure(values=volume_options)
        from_option.set('Liter')
        to_option.set('Milliliter')
    elif variable1.get() == 'Temperature':  # Added
condition for temperature options
        from_option.configure(values=temperature_options)
        to_option.configure(values=temperature_options)
        from_option.set('Celsius')
        to_option.set('Fahrenheit')

variable1.trace("w", update_options)
app.mainloop()
```

REFERENCES

Converter using tkinter:

https://thecleverprogrammer.com/2020/11/28/weight-converter-gui-with-python/?fbclid=IwZXh0bgNhZW0CMTAAAR15IJSA3mrfKT6TP9P8Zld26TcwYmna2beYNt3qVPCSMeBaVjui4EHkMvw_aem_ATw1mH-MiL6GOPZ0opL0zI9kJA1G0tU9b8Ywgph1yB_L2I0JNwn1m3NMOJjGRbhhxKcUxKm7vRaXghWtdKt4F5lf

How to use customtkinter:

https://www.youtube.com/watch?v=iM3kjbbKHQU&list=PLwmJKHBfA6Nt6w2MrPilWKzFb45Y1VGUF&index=5