

CURRENCY CONVERTER PROJECT WITH GUI

by;

- IU2141050005-NARESH BARIYA
- IU2141050029-DEV DARJI

STEPS TO BUILD THE PYTHON PROJECT ON CURRENCY CONVERTER

- 1.Import the necessary Libraries
- 2.Creating a GUI window
- 3.Currency Converter Class
- 4.UI for Currency Converter
- 5.Main Function

1.IMPORT THE LIBRARIES

- Tkinter-It is the inbuilt python module that is used to create GUI applications. To install Tkinter, go to command prompt and type pip install tkinter
- `import tkinter as tk`
- `from currency_converter import CurrencyConverter`

2. CREATING A GUI WINDOW AND HEADING FOR IT

- `window = tk.Tk()` This line of code automatically creates a GUI window with a title bar
- We can change the title using title of root window using `window.title(" ")`
- `window = tk.Tk()`
- `window.geometry("800x360")`
- `window.title("Currency Converter")`

3.CURRENCY CONVERTER CLASS

- Create a ' CurrencyConverter 'object named 'c' from the import class
- # Create a CurrencyConverter object
- `c = CurrencyConverter()`

4. UI FOR PYTHON CURRENCY CONVERTER PROJECT

- Define label for various parts of the interface(title ,input label and result)
- Create entry widgets for user input(amount and currencies)
- Add a button for a triggering the conversion
- Place the widgets in the window using 'place()'
- Add Dropdown menu for source currency
- **# Labels**
 - `l1 = tk.Label(window, text="Currency Converter", font="Times 25 bold")`
 - `l1.place(x=100, y=30)`
 - `l2 = tk.Label(window, text="Enter amount here:", font="Times 18 bold")`
 - `l2.place(x=50, y=80)`
 - `l3 = tk.Label(window, text="Select Source Currency:", font="Times 18 bold")`
 - `l3.place(x=50, y=130)`
 - `l4 = tk.Label(window, text="Select Target Currency:", font="Times 18 bold")`
 - `l4.place(x=50, y=180)`

CONTINUE...

- # Entry widget for amount

- `e1 = tk.Entry(window)`
- `e1.place(x=300, y=90)`

- # Button

- `b1 = tk.Button(window, text="Convert", command=clicked)`
- `b1.place(x=230, y=240)`
- `b2 = tk.Button(window, text="Clear", command=clear)`
- `b2.place(x=320, y=240)`

CONTINUE.....

- # Dropdown menu for source currency


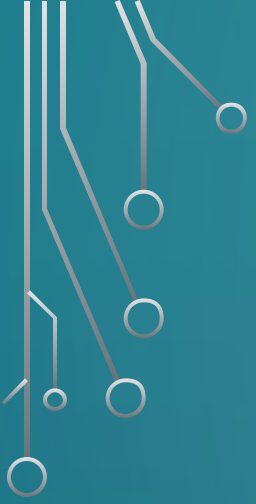
- `source_currency_list = ['USD', 'EUR', 'JPY', 'GBP', 'AUD', 'INR']`
- `var_source_currency = tk.StringVar()`
- `var_source_currency.set(source_currency_list[0])`
- `source_currency_menu = tk.OptionMenu(window, var_source_currency, *source_currency_list)`
- `source_currency_menu.place(x=300, y=140)`

- # Dropdown menu for target currency

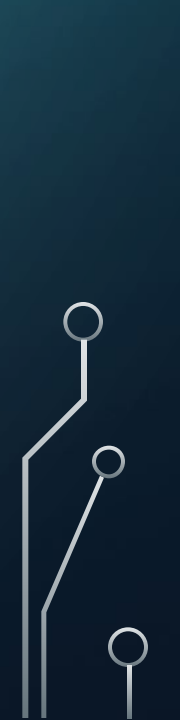

- `target_currency_list = ['USD', 'EUR', 'JPY', 'GBP', 'AUD', 'INR']`
- `var_target_currency = tk.StringVar()`
- `var_target_currency.set(target_currency_list[0])`
- `target_currency_menu = tk.OptionMenu(window, var_target_currency, *target_currency_list)`
- `target_currency_menu.place(x=300, y=190)`

5.MAIN FUNCTION

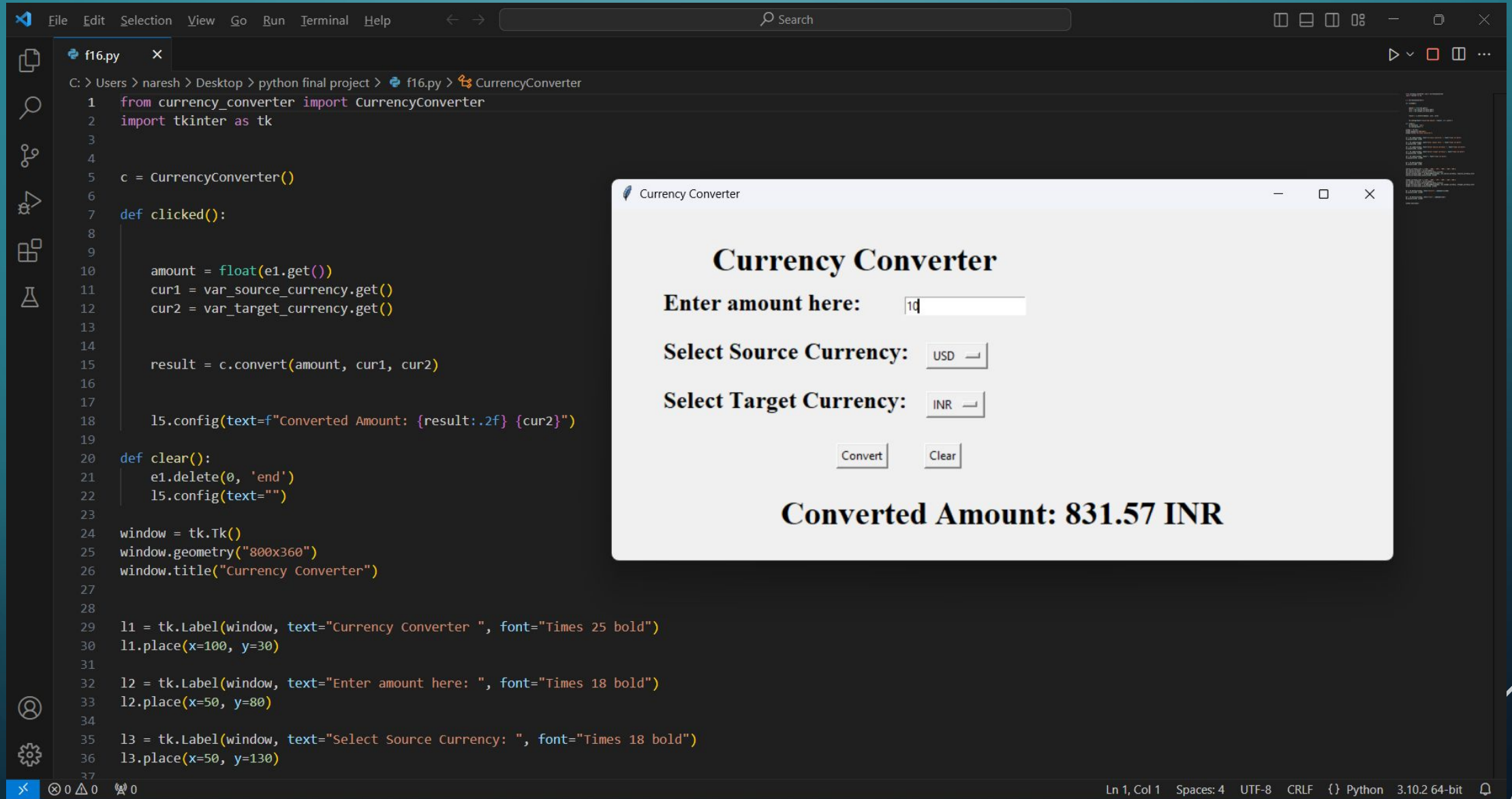
- Define a clicked() function that gets called when the "Convert" button is clicked.
- Inside clicked(), retrieve the user input values from entry widgets, perform the currency conversion using the CurrencyConverter object, and update the result label with the converted amount.
- Finally, start the main event loop using window.mainloop(), which keeps the GUI application running and responsive to user interactions



```
•
• def clicked():
•     # Retrieve values from entry widgets
•     amount = float(e1.get())
•     cur1 = var_source_currency.get()
•     cur2 = var_target_currency.get()
•
•     # Perform the currency conversion
•     result = c.convert(amount, cur1, cur2)
•
•     # Update the result label
•     l5.config(text=f"Converted Amount: {result:.2f}
{cur2}")
```



CURRENCY CONVERTER OUTPUT



The image displays a Python script in a code editor, titled `f16.py`, which implements a currency converter. The script imports the `CurrencyConverter` class and `tkinter` as `tk`. It creates an instance `c = CurrencyConverter()` and defines a `clicked()` function that takes the input amount, source currency, and target currency, converts them, and updates the GUI. A `clear()` function is also defined to reset the input fields. The GUI is created using `tk.Tk()` with a title "Currency Converter" and a geometry of "800x360". It includes labels for the title, input amount, source currency, and target currency, along with "Convert" and "Clear" buttons. The output of the conversion is displayed as "Converted Amount: 831.57 INR".

```
1 from currency_converter import CurrencyConverter
2 import tkinter as tk
3
4
5 c = CurrencyConverter()
6
7 def clicked():
8
9     amount = float(e1.get())
10    cur1 = var_source_currency.get()
11    cur2 = var_target_currency.get()
12
13
14    result = c.convert(amount, cur1, cur2)
15
16
17    l5.config(text=f"Converted Amount: {result:.2f} {cur2}")
18
19
20 def clear():
21    e1.delete(0, 'end')
22    l5.config(text="")
23
24 window = tk.Tk()
25 window.geometry("800x360")
26 window.title("Currency Converter")
27
28
29 l1 = tk.Label(window, text="Currency Converter ", font="Times 25 bold")
30 l1.place(x=100, y=30)
31
32 l2 = tk.Label(window, text="Enter amount here: ", font="Times 18 bold")
33 l2.place(x=50, y=80)
34
35 l3 = tk.Label(window, text="Select Source Currency: ", font="Times 18 bold")
36 l3.place(x=50, y=130)
```

The graphical output window, titled "Currency Converter", shows the following interface:

- Enter amount here:** 10
- Select Source Currency:** USD
- Select Target Currency:** INR
- Convert** and **Clear** buttons
- Converted Amount: 831.57 INR**

SUMMARY

- With this project in python, we have successfully developed and executed the real time Currency Converter Project.
- We used tkinter and currency_converter modules to develop our project.
- Executing different functions and using many widgets and using labels us to develop our python programming skills

The background is a teal-to-dark-blue gradient. In the corners, there are decorative white line art elements resembling circuit boards or neural networks, with lines and small circles.

THANK YOU