Python Mini Project

```
import tkinter.messagebox
from tkinter import *
from tkinter import ttk



def RealTimeCurrencyConversion():
    from forex_python.converter import CurrencyRates


    c = CurrencyRates()


    from_currency = fcurrency.get()
    to_currency = tcurrency.get()


    if amount_entry.get() == "":
        tkinter.messagebox.showinfo(
            "Error !!", "Amount Not Entered.\n Please Enter a valid amount.")


    elif from_currency == "" or to_currency == "":
        tkinter.messagebox.showinfo("Error !!",
                                    "Currency Not Selected.\n Please select FROM and TO Currency from
menu.")


    else:
        converted_entry.delete(0, END)
        new_amt = c.convert(from_currency, to_currency,
                            float(amount_entry.get()))
        new_amount = float("{:.4f}".format(new_amt))
```

```
        converted_entry.insert(0, str(new_amount))


        # Connecting Database
        def store():


            import mysql.connector


            am = amount.get()
            fc = fcurrency.get()
            tc = tcurrency.get()
            ca = converted.get()


            mydb = mysql.connector.connect(host="localhost", user="root",
password="@rshadK786",
                                           database="currencyconverter")
            cursor = mydb.cursor()


            query = "INSERT INTO
conversion(Amount,From_Currency,To_Currency,Converted_Amount)
VALUES(%s,%s,%s,%s)"
            vals = (am, fc, tc, ca)
            cursor.execute(query, vals)
            mydb.commit()
        store()



def clear_all():
    converted_entry.delete(0, END)
    amount_entry.delete(0, END)
    fcurrency.set("")
```

```
    tcurrency.set("")


def convert():
    home.pack_forget()
    f1.pack(pady=25)
    f2.pack()


def home1():
    f1.pack_forget()
    f2.pack_forget()
    home.pack()


CurrenyCode_list = ["INR", "USD", "CAD", "CNY", "DKK", "EUR"]

root = Tk()
root.title("Currency Conversion System")
# Size of the GUI window
root.geometry("800x600")
# root.configure(bg="grey")
root.minsize(800, 600)
root.maxsize(1366, 768)
root.iconbitmap("Google-Noto-Emoji-Objects-62885-currency-exchange.ico")

# Home Frame
home = Frame(root)
home.pack()
```

```
# Background Imagge
image = PhotoImage(file="background.png")
canvas1 = Canvas(home, width="800", height="600")
canvas1.pack(fill="both", expand=True)
# Display image
canvas1.create_image(0, 0, image=image,
                     anchor="nw")

# Heading
f1 = Frame(root)
# f1.pack(pady=25)
Label(f1, text="Currency Converter", font="consolas 30 underline bold").pack()

# Labels
f2 = Frame(root)
# f2.pack()
Label(f2, text="Amount: ", font="consolas 20 bold").grid(
   row=3, column=2, ipady=10)
Label(f2, text="From Currency: ", font="consolas 20 bold").grid(
   row=4, column=2, ipady=10)
Label(f2, text="To Currency: ", font="consolas 20 bold").grid(
   row=5, column=2, ipady=10)
Label(f2, text="Converted Amount: ", font="consolas 20 bold").grid(
   row=6, column=2, ipady=10)

# Datatypes
amount = StringVar()
converted = StringVar()
```

```
fcurrency = StringVar()

tcurrency = StringVar()


# Entry Widgets

amount_entry = Entry(f2, textvariable=amount)

amount_entry.grid(row=3, column=3)


converted_entry = Entry(f2, textvariable=converted)

converted_entry.grid(row=6, column=3)


# OptionMenu

FromCurrency_option = OptionMenu(f2, fcurrency, *CurrenyCode_list)

FromCurrency_option.grid(row=4, column=3, ipadx=40)


ToCurrency_option = OptionMenu(f2, tcurrency, *CurrenyCode_list)

ToCurrency_option.grid(row=5, column=3, ipadx=40)


# Menu

mymenu = Menu(root)

mymenu.add_command(label="Home", command=home1)

mymenu.add_command(label="Convert", command=convert)

root.configure(menu=mymenu)


# Buttons

Button(f2, text="Convert", command=RealTimeCurrencyConversion, padx=25, pady=20,
fg="white", bg="blue",

    font="consalas 12 bold",

    relief="raised", bd=5).grid(row=7, column=2, pady=15)

Button(f2, text="Clear All", command=clear_all, padx=25, pady=20, fg="white", bg="red",
font="consalas 12 bold",
```

```
        relief="raised", bd=5).grid(row=7, column=3, pady=15)


root.mainloop()
```