

PROJECT DEVELOPMENT PHASE
Sprint - 1
Code for Calculates the Nutrition Values

Date	14 November 2022
Team Id	PNT2022TMID03936
Project Name	AI- powered Nutrition Analyzer and Fitness Enthusiasts

#App for nutrition values

```
from tkinter import *  
from tkinter import messagebox
```

```
class App(Tk):
```

```
    def __init__(self):  
        Tk._init_(self)  
        self._frame = None  
        self.title("App calculating nutrition values")  
        self.switch(Menu)  
        self.geometry('350x350')  
        self.config(bg = "black")
```

```
    def switch(self, frame_class):  
        """Destroys current frame and replaces it with a chosen by the user"""  
        new_frame = frame_class(self)  
        if self._frame is not None:  
            self._frame.destroy()  
        self._frame = new_frame  
        self._frame.pack()
```

```
class Menu(Frame):
```

```
    """Main menu"""  
    def __init__(self, master):  
        Frame._init_(self, master)  
        self.config(bg = "black")
```

```

"""Frame widgets"""
label = Label(self, text = "Welcome in a nutrition calculator!\n Choose an option."\
, bg = "black", fg = "white")
label.pack()
button = Button(self, text = "Calculator", width = 20, command = lambda:
master.switch(Calculator))
button.pack(padx = 10, pady = 10)
button2 = Button(self, text = "Add a product", width = 20, command = lambda:
master.switch(File_Write))
button2.pack()
button3 = Button(self, text = "Exit", width = 20, command = self.close)
button3.pack(padx = 10, pady = 10)

def close(self):
    """Close the app"""
    self.destroy()
    exit()

```

```

class Calculator(Frame):
    """Writing nutritional values of the user defined food"""
    def __init__(self, master):
        Frame.__init__(self, master)
        self.config(bg = "black")

    def on_click():
        """Checking data and writing the results"""
        product = entryProduct.get()
        gram = entryGram.get()
        output.delete(0.0, END)

        Error = False
        try:
            gram = int(entryGram.get())
        except:
            Error = True
        try:
            x = int(product)
            Error = True
        except:

```

```

        pass
    if Error == True:
        messagebox.showerror("Error", "Please enter correct data!")
    else:
        functions.file_open()
        output.insert(END, functions.result(product, gram))

"""Frame widgets"""
label = Label(self, text = "Enter a product that you ate.", bg = "black", fg = "white")
label.pack()
# user input, product
label2 = Label(self, text = "Name: ", bg = "black", fg = "white")
label2.pack()
entryProduct = Entry(self, width = 20, bg = "white")entryProduct.pack()
# user input, amount
label3 = Label(self, text = "Amount: ", bg = "black", fg = "white")
label3.pack()
entryGram = Entry(self, width = 20, bg = "white")
entryGram.pack()
# submit
submit = Button(self, text = "Submit", width = 8, command = on_click)
submit.pack(padx = 10, pady = 10)
# output
label4 = Label(self, text = "These are the nutrinion values:", bg = "black", fg = "white")
label4.pack()
output = Text(self, width = 20, height = 6, wrap = WORD, bg = "white")
output.pack()
#going back to menu
self.button = Button(self, text = "Back", width = 8, command = lambda:
master.switch(Menu))
self.button.pack(padx = 10, pady = 10)

class File_Write(Frame):
    """User can add new new products and their values""" def_
    init_(self, master):
        Frame._init_(self, master)
        self.config(bg = "black")

    def validate():

```

```

"""Checks if the user inputs correct data"""
def write(name, kcal, protein, carb, fat):
    """Writes to file"""
    file = open("Products.txt", "a")
    productValue = "%s,%s:%s:%s:%s" % (name, kcal, protein, carb, fat)
    file.write("\n" + productValue)
    file.close()
    #Emptying inputs
    nameEntry.delete(0, END)
    kcalEntry.delete(0, END)
    proteinEntry.delete(0, END)
    carbEntry.delete(0, END)
    fatEntry.delete(0, END)

error = False
# checking if kcal, protein, carb and fat are integers and productName is a string
try:
    name = int(nameEntry.get())
    error = True
except:
    name = nameEntry.get()
try:
    kcal = int(kcalEntry.get())
    protein = int(proteinEntry.get())
    carb = int(carbEntry.get())
    fat = int(fatEntry.get())
except:
    error = True
if error == True:
    messagebox.showerror("Error", "Please enter correct data!")
else:
    #writing to a file
    write(name, kcal, protein, carb, fat)

"""Frame widgets"""
label = Label(self, text = "Enter the product name and its nutritional \
    values per 100 gram", bg = "black", fg = "white")
label.pack()
label1 = Label(self, text = "Name:", bg = "black", fg = "white")
label1.pack()
nameEntry = Entry(self, width = 20, bg = "white")

```

```
nameEntry.pack()
```

```
label2 = Label(self, text = "Calories:", bg = "black", fg = "white")
```

```
label2.pack()
```

```
kcalEntry = Entry(self, width = 20, bg = "white")
```

```
kcalEntry.pack()
```

```
label3 = Label(self, text = "Protein:", bg = "black", fg = "white")
```

```
label3.pack()
```

```
proteinEntry = Entry(self, width = 20, bg = "white")
```

```
proteinEntry.pack()
```

```
label4 = Label(self, text = "Carbs:", bg = "black", fg = "white")
```

```
label4.pack()
```

```
carbEntry = Entry(self, width = 20, bg = "white")
```

```
carbEntry.pack()
```

```
label5 = Label(self, text = "Fat:", bg = "black", fg = "white")label5.pack()
```

```
fatEntry = Entry(self, width = 20, bg = "white")
```

```
fatEntry.pack()
```

```
submit = Button(self, text = "Submit", width = 8, command = validate)
```

```
submit.pack(padx = 10, pady = 10)
```

```
button3 = Button(self, text = "Back", width = 20, command = lambda: master.switch(Menu))
```

```
button3.pack(padx = 10, pady = 10)
```

```
if __name__ == "__main__":app
```

```
    = App() app.mainloop()
```