SAAD'S SMART ORIENTATION APP BREAKDOWN

Part By Part Detailed Breakdown

"Made by Saad Adnane"

DESIGN

- #!/usr/bin/env python3
- # coding: utf-8
- Upwards I'm just telling your PC to run this as a Python code "the first line isn't needed in Windows"
- In the second line I'm adding support to imogies, french accents and arabic letters "in case I needed them"
- import tkinter as tk
- from tkinter import ttk, messagebox
- first line is importing a library that I need and aliasing it as tk
- And the second I'm importing ttk "Themed tkinter"

- root = tk.Tk()
- root.title("Saad's Orientation Assistant")
- Creating main frame and title

Downwards I'm creating labels and entries "simply design" and tricking tkinter so I can use ".grid" and ".pack" in the same window

```
MainLabel1 = tk.Label(root, text = "Welcome to Saad's Orientation app for 2AC!", font=("Arial", 16, "bold") ,foreground="Blue")
MainLabel1.pack()

MainLabel2 = tk.Label(root, text = "Write down your marks to see which orientation is best for you", font=("Arial", 12), pady = 10)
MainLabel2.pack()


Frame = ttk.Frame(root)
Frame.pack(pady = 20)


Label1 = ttk.Label(Frame, text = "Physics:", font = ("Arial", 10))
Label1.grid(row = 1, column = 0, sticky = "w", padx = 10, pady = 10)

Entry1 = ttk.Entry(Frame)
Entry1.grid(row = 1, column = 1, sticky = "w", padx = 10, pady = 10)


Label2 = ttk.Label(Frame, text = "Hist.Geo:", font = ("Arial", 10))
Label2.grid(row = 2, column = 0, sticky = "w", padx = 10, pady = 10)

Entry2 = ttk.Entry(Frame)
Entry2.grid(row = 2, column = 1, sticky = "w", padx = 10, pady = 10)
```

- Label3 = ttk.Label(Frame, text = "ICT:", font = ("Arial", 10))
- Label3.grid(row = 3, column = 0, sticky = "w", padx = 10, pady = 10)

- Entry3 = ttk.Entry(Frame)
- Entry3.grid(row = 3, column = 1, sticky = "w", padx = 10, pady = 10)

- Label4 = ttk.Label(Frame, text = "Edu.Islamique", font = ("Arial", 10))
- Label4.grid(row = 4, column = 0, sticky = "w", padx = 10, pady = 10)

- Entry4 = ttk.Entry(Frame)
- Entry4.grid(row = 4, column = 1, sticky = "w", padx = 10, pady = 10)

- Label5 = ttk.Label(Frame, text = "English", font = ("Arial", 10))
- Label5.grid(row = 5, column = 0, sticky = "w", padx = 10, pady = 10)

- Entry5 = ttk.Entry(Frame)
- Entry5.grid(row = 5, column = 1, sticky = "w", padx = 10, pady = 10)

- Label6 = ttk.Label(Frame, text = "Arabic", font = ("Arial", 10))
- Label6.grid(row = 6, column = 0, sticky = "w", padx = 10, pady = 10)

- Entry6 = ttk.Entry(Frame)
- Entry6.grid(row = 6, column = 1, sticky = "w", padx = 10, pady = 10)

- Label7 = ttk.Label(Frame, text = "French", font = ("Arial", 10))
- Label7.grid(row = 7, column = 0, sticky = "w", padx = 10, pady = 10)

- Entry7 = ttk.Entry(Frame)
- Entry7.grid(row = 7, column = 1, sticky = "w", padx = 10, pady = 10)

- Label8 = ttk.Label(Frame, text = "Maths", font = ("Arial", 10))
- Label8.grid(row = 8, column = 0, sticky = "w", padx = 10, pady = 10)

- Entry8 = ttk.Entry(Frame)
- Entry8.grid(row = 8, column = 1, sticky = "w", padx = 10, pady = 10)

- Label9 = ttk.Label(Frame, text = "SVT", font = ("Arial", 10))
- Label9.grid(row = 9, column = 0, sticky = "w", padx = 10, pady = 10)

- Entry9 = ttk.Entry(Frame)
- Entry9.grid(row = 9, column = 1, sticky = "w", padx = 10, pady = 10)

THE CORE OF THE PROGRAM

- Saving mechanism; grabs grades and the best orientation and put'em in a file called: Orientation_Result.txt
- def save_to_file(result, marks):
- import os

- file_path = os.path.join(
- os.path.dirname(os.path.abspath(__file__)),
- "Orientation_Result.txt"
-)

- with open(file_path, "w", encoding="utf-8") as file:
- file.write("=== Here Are Your Marks And Your Best Orientation! ===\n")
- for subject, value in marks.items():
- file.write(f"{subject}: {value}\n")
- file.write(f"Best Orientation: {result}\n\n")

- Initializing variables and checking that all inputs contain numbers after clicking the button
- If there is a problem an error will be thrown to the user

```
def Submit():
    global Physics, HistGeo, ICT, EduIslamique, English, Arabic, French, Maths, SVT
    try:
        Physics          = float (Entry1.get())
        HistGeo           = float (Entry2.get())
        ICT              = float (Entry3.get())
        EduIslamique      = float (Entry4.get())
        English          = float (Entry5.get())
        Arabic           = float (Entry6.get())
        French           = float (Entry7.get())
        Maths            = float (Entry8.get())
        SVT              = float (Entry9.get())
    except ValueError:
        messagebox.showerror("Input Error", "Please fill all fields with numbers only.")


        return
```

- Initializing variables containing every orientaton's marks, calculating the one with most marks, putting in in a variable called "Best" and calculating total orientations scores "For Percentage"
- AuthenticEducation = (Physics * 0) + (HistGeo * 3) + (ICT * 2) + (EduIslamique * 4) + (English * 2) + (Arabic * 4) + (French * 3) + (Maths * 2) + (SVT * 2)
- ArtsAndHumanities = (Physics * 0) + (HistGeo * 4) + (ICT * 2) + (EduIslamique * 0) + (English * 3) + (Arabic * 4) + (French * 4) + (Maths * 2) + (SVT * 2)
- ScientificTrunk = (Physics * 4) + (HistGeo * 2) + (ICT * 2) + (EduIslamique * 0) + (English * 3) + (Arabic * 2) + (French * 3) + (Maths * 4) + (SVT * 4)
- TechnologicalStump = (Physics * 4) + (HistGeo * 2) + (ICT * 3) + (EduIslamique * 0) + (English * 3) + (Arabic * 2) + (French * 3) + (Maths * 4) + (SVT * 0)
- Best = max(AuthenticEducation, ArtsAndHumanities, ScientificTrunk, TechnologicalStump)
- TotalScore = AuthenticEducation + ArtsAndHumanities + ScientificTrunk + TechnologicalStump

```
Outputing to the user based on the best Orientation
if (Best == AuthenticEducation):
        best_orientation = "Authentic Education"

    elif(Best == ArtsAndHumanities):
        best_orientation = "Arts And Humanities"

    elif(Best == ScientificTrunk):
        best_orientation = "Scientific Trunk"

    elif(Best == TechnologicalStump):
        best_orientation = "Technological Stump"

    messagebox.showinfo(
        "Analysis Results",
        f"The Best Orientation For You is: {best_orientation}\n\n"
        f"Authentic Education: {AuthenticEducationPercent:.0f}%\n"
        f"Arts And Humanities: {ArtsAndHumanitiesPercent:.0f}%\n"
        f"Scientific Trunk: {ScientificTrunkPercent:.0f}%\n"
        f"Technological Stump: {TechnologicalStumpPercent:.0f}%\n"
        )
```

- Creating a variable of type "dictionary" to store the name of subject plus its value "the mark you typed"
- marks = {
- "Physics": Physics,
- "Hist.Geo": HistGeo,
- "ICT": ICT,
- "Edu.Islamique": EduIslamique,
- "English": English,
- "Arabic": Arabic,
- "French": French,
- "Maths": Maths,
- "SVT": SVT
- }
- Calling the function we set in Slide 8
- save_to_file(best_orientation, marks)

- Creating a "Submit" button
- SubmitButton = tk.Button(Frame, text = "Submit", bg = "gray", command = Submit)
- SubmitButton.grid(row = 10, column = 0, columnspan = 2, pady = 10)
- Keeping the app running over and over until the user exits
- root.mainloop()