

**Premier league Project**

**Team members**.

1. Mohamed Sabry Mahmoud Shehab Eldean (Team Leader).

2. Mohamed Ahmed Mohamed Emam.

3. Mohamed Tarek Amin.

4. Fatima Ibrahim Kamel.

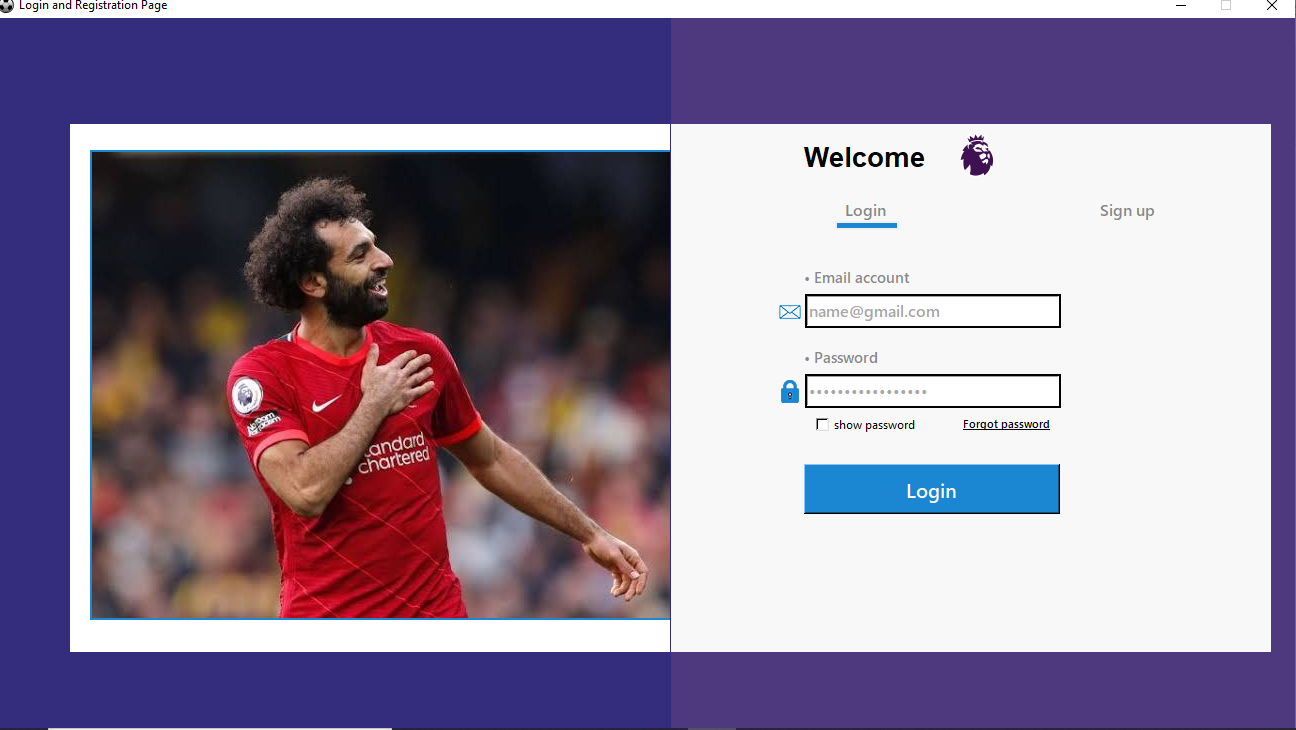
5. Leqaa Mohamed Abdallah.

6. Mohamed Bahgat Abdel Khaleq.

Project Description

This project uses the user to know the dates of football matches of the English Premier League and know the football news and know information about the players of the English league such as (age / club / nationality / number of games played / number of scorers recorded .... etc) and it is possible through the supervisor to add the matches that were played in the league.

➢Welcome window.

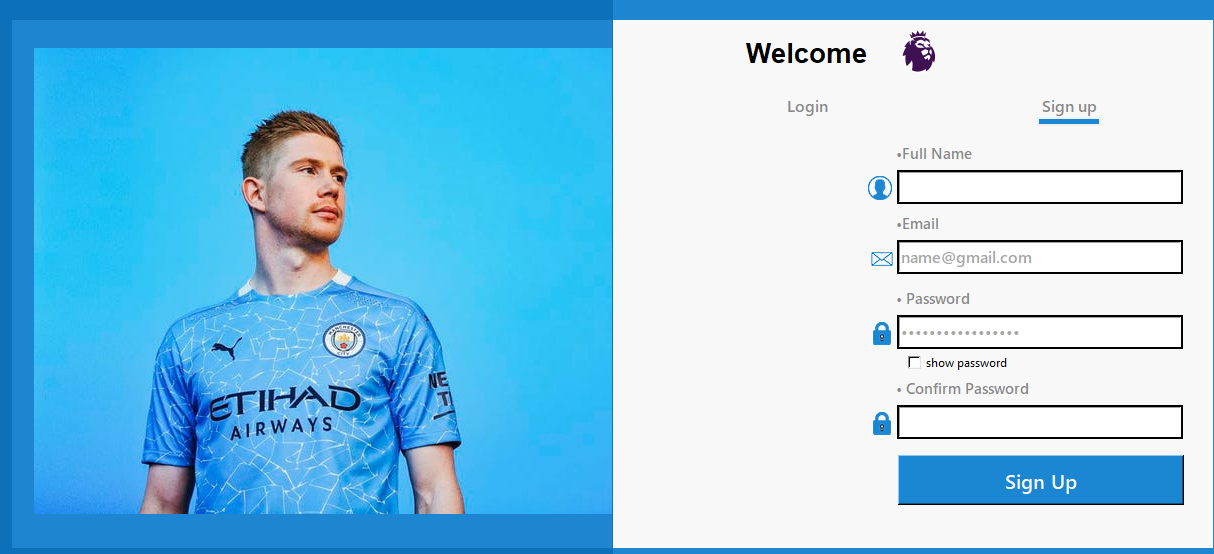
When the project starts, a "welcome" window appears and the user is asked to enter their email and password to be able to log in. There is a login button that takes you to the next window if the data entered is correct it will show him a message confirming that the input is correct and if the user enters the wrong data will appear to him a message indicating that I am the user may enter the data wrongly.

First the screen of the window is defined as "Four and in Lindo. Info \_Children() " and then both Roconfiger and Colonenfiguri are selected "Lindo Roconfiger(0, IT=1) Lindo Colomonfigiuri(0, IT=1)" and State and Resizabel" Lindo.State ('Zomed') Lindo. eesable(0, 0) " and screen dimensions " Lindo.geometric ('1300 g800') "and the title name of the page is "Lindo.Tetley ('Login and Registry Page')" and the image used is defined on the page "Icon = PhotoMagic (File='ImageBeach-Icon.ping')".

The type and background color "LoginPage=Frame (window, bg='white')"is specified "The image that will be placed as the background is specified "confirmation = frame(window)".

➢sign up.

And if he does not have an account, he can click on sign up, a window appears and the required data is entered such as name, password, e-mail and password confirmation, and then a message is received on the e-mail that enables you to enter the application by adding the new user to the database.

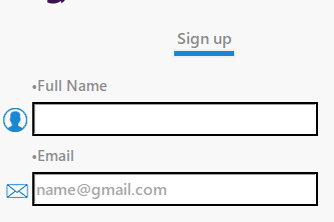


The mechanism of creating the background show "frame. tkraise ()" and Email, Full Name, Password and Confirm Password are taken from a new user and saved to the database “Email = String Var()  
Full Name = String Var()  
Password = String Var()  
Confirm Password = String Var() “.

4 frames are made to form the shape of the window "design\_frame1" and its size is limited to "width=115, height=50" and then the amount of highlighting is specified and the width of the border is specified "highlight thickness=0, border width=0" and then its location is determined on the screen "design\_frame1.place(x=0, y=0)" and these steps are applied 3 more times with the location of the frame changing according to the use.

"Email \_entry" is defined as the place where the user will enter to type his own email and the type and size of the font written in it is "font=("yu gothic ui semibold", 12)" and the text specified in it is the email "text variable=Email" and the font type and font size

"email\_entry.place(x=134, y=170, width=256, height=34)"

are specified and then the type of lighting is specified "highlightbackground="black", highlightcolor="black""

And a default value of the tilt

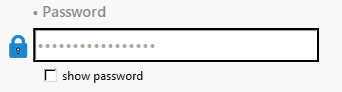
"email\_entry.insert(0,"name@gmail.com")"

Is then defined as "email \_label" to

Be placed next to it as belonging

To the user's email "design\_frame4, text='• Email" and then its location is specified on the original frame "email\_ label. place(x=130, y=140)".

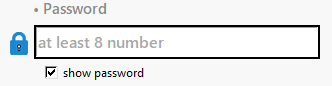
"password\_entry1" is defined as the place where the user will enter to type his password and the type and size of the font written in it is "font=("yu gothic ui semibold", 12)" and the text specified in it is the password "text variable=Password" and the font type and font size "x=134, y=250, width=256, height=34)" are specified and then the type of lighting is determined "highlight background="black", highlight color="black""

And the number of characters is

Specified for the password

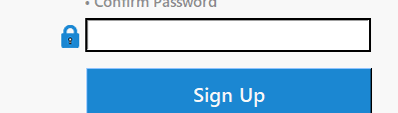
"password\_entry1.insert(0,"at least 8 number")"

And then "password \_label" is defined to be placed next to it as being specific to the user's password "4, text='• Password'" and then located on the original frame "password\_ label.place(x=130, y=220)".

def password\_command():  
 if password\_entry1.cget('show') == '•':  
 password\_entry1.config(show='')  
 else:  
 password\_entry1.config(show='•')

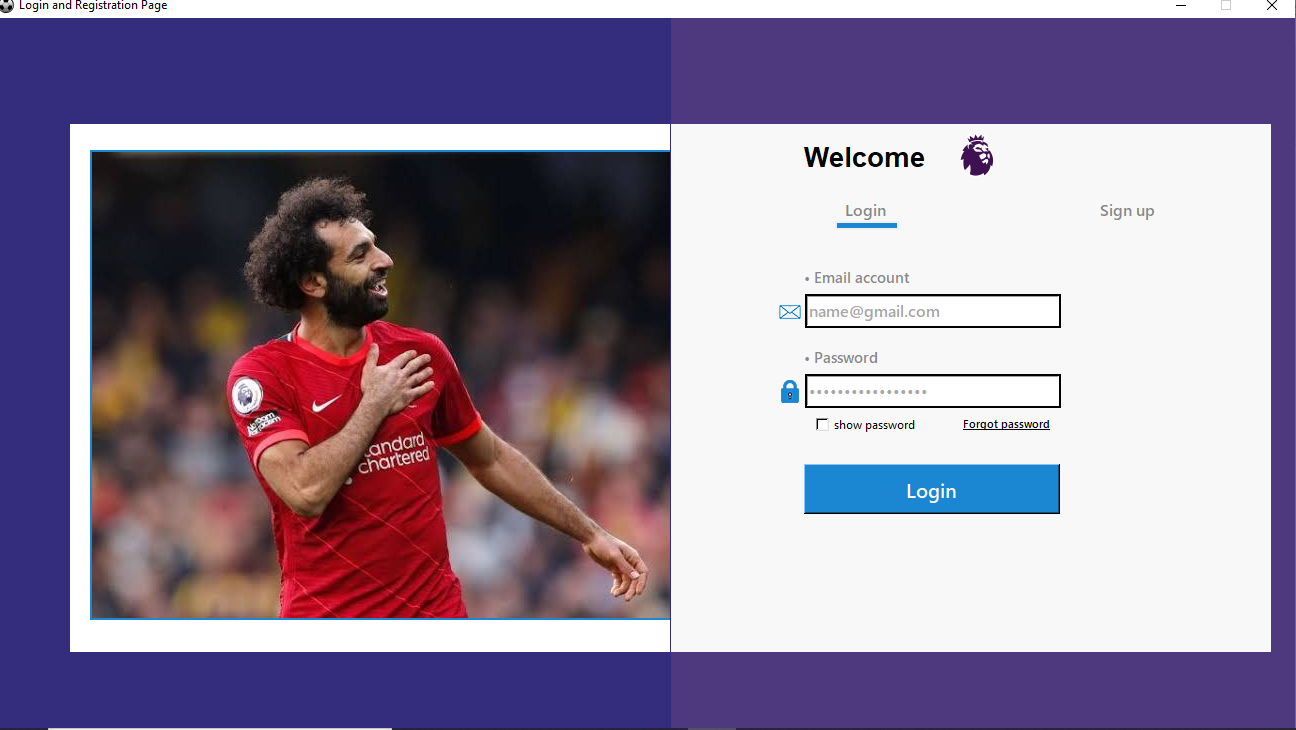
And this function is made so that when the user presses the word "show" it shows him the password in the form of numbers and letters but how if he presses it the password is written in the form of "\*\*\*".

A button is made through which the PIN code will be entered by the user "check button" and placed in the fourth frame "design \_frame4" and then the entered text "text='show password'" is selected and then the command is placed to confirm the operation "command=password \_command" and then its place is specified on the window" check Button .place(x=140, y=288)".

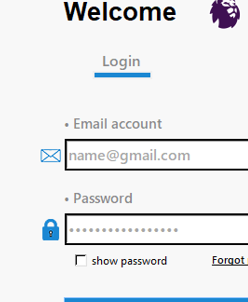
Then the button that the user presses is made to confirm the operation "Sign Up \_button" and then the name of the text is placed and the font type and size are specified

"Text='Sign up', font=("yu gothic ui bold", 12)"

And then it is linked to the command

"Command=lambda: show \_frame(Registration Page)" to confirm the process and then the button is located on the window "Sign Up\_ button. place(x=1100, y=175)".

Then "welcome \_label" is made through which the user will be able to access the program and then the text, its type and size are placed "text='Welcome', font=('Arial', 20, 'bold')" and then the text is placed on the window "welcome \_label .place(x=130, y=15)" and then the button is made through which the confirmation process "login\_ button" is determined and the type and size of the font used is specified and the text written "text='Login', is selected. font=("yu gothic ui bold", 12)" and then the button is located on the window "loginBtn1.place(x=133, y=340, width=256, height=50)" and then the command "command=lambda: login()" is specified through which the validity of the slope and password "command=lambda: login()" is checked.

Then the image used is defined on the window "email \_icon = Image .Open('imagesemail-icon.png')" and then the "email Icon\_ label" through which the image will be displayed is defined and then the image is marked with "design\_frame4, image=photo" and then its place is determined on the frame.

Then the icon is placed both

The password, username and

Email "password \_icon and picture \_icon"

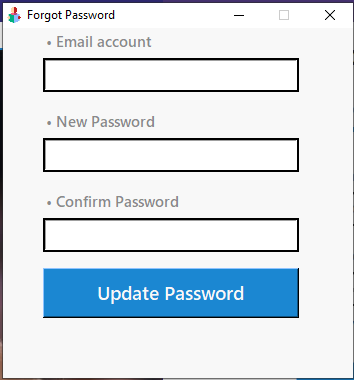
In the same way with different

Places on the window and the written text.

Then a function is made through which a message is sent to the user if his data is correct and located inside the database shows him a message that the registration was successful "message box .show Info("Success", 'Logged in Successfully.')" and if the data is not registered correctly, a message is sent to him with the failure of the registration "else:

Message box .show error("Failed", "Wrong Login details, please try again.")".

|  |  |
| --- | --- |
| If the user enters the wrong data | If the user enters the correct data |
| C:\Users\user\Desktop\745.PNG | C:\Users\user\Desktop\df.PNG |

A function is made inside it

"def forgot\_password():" which will help

The user how he forgot the

password"win.title('Forgot Password')"

Will ask him to fill in his

Inclination once and then put

A new password soThat he can

Access the site "Win.geometry (f'{window\_width}x{window\_height}+{position\_right}+{position\_top}')"

Position \_top = int(screen \_height / 4 – window \_height / 4)

Position \_right = int(screen\_ width / 2 – window \_width / 2)" Then the window for changing the password Appears is determined " window\_ width = 350

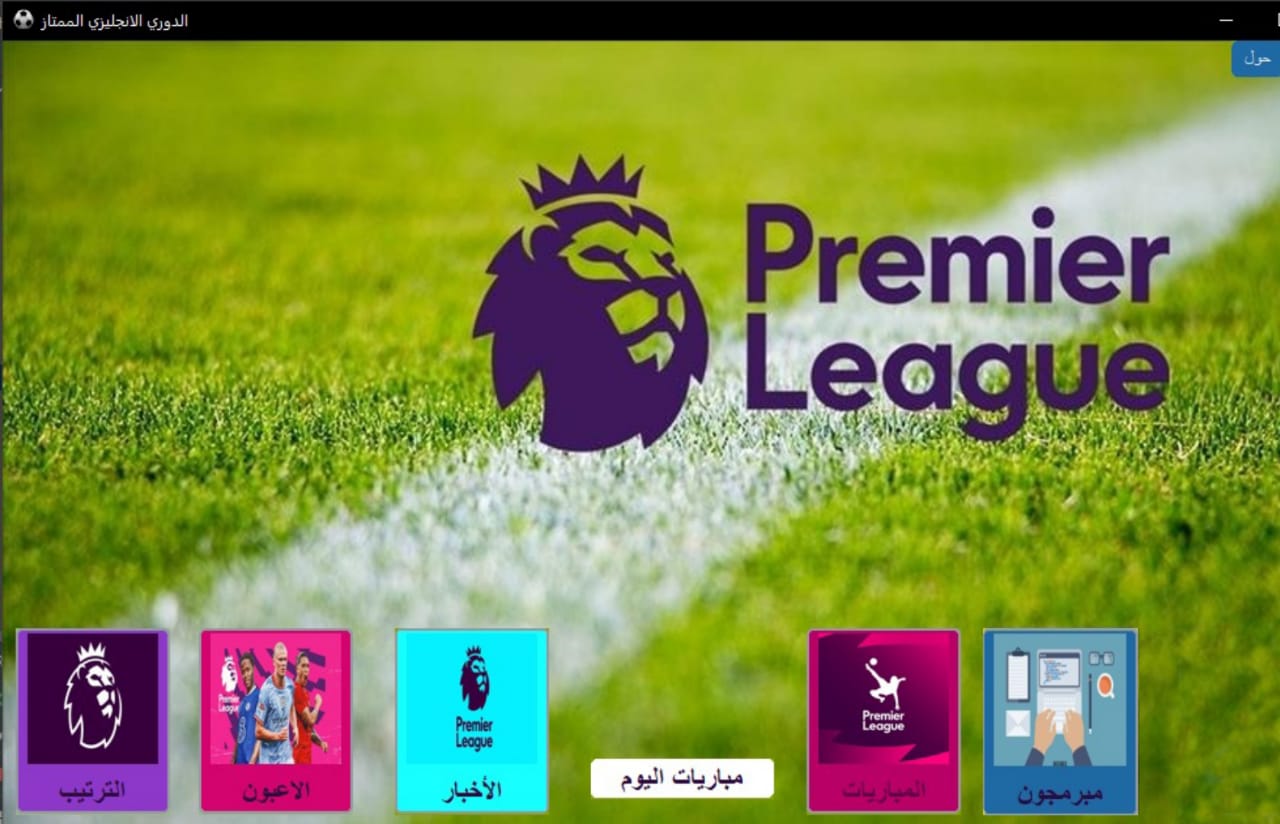
Window \_height = 350 Screen \_width = win .winfo \_screen width()

Screen \_height = win. winfo\_ screen height()".

If the operation is successful, a confirmation message is sent "message box .show info('Congrats', 'Password changed successfully')" and if the operation is not successful, it sends you a message retrying "message box .show error('Error!', 'Passwords didn't match)".

Then the "email\_entry2" is defined by which the user's email will be written and then we specify the size and type of font used " font=("yu gothic ui semibold", 12)" and then put the command "textvariable=email" through which the validity of the user's inclination will be known and then located on the window " email\_entry2.place(x=40, y=30, width=256, height=34)".

|  |  |
| --- | --- |
| The page that will appear to the average user | The page that will appear to the admin |
|  | D:\project\p\WhatsApp Image 2022-08-17 at 10.07.18 PM.jpeg |



➢Main window.

This project consists of 9 basic windows in the main interface and they are (about / reset / ranking / players / news / matches today / games / add games / programmers) and each window has a specific mechanism of work that we will learn about in detail.

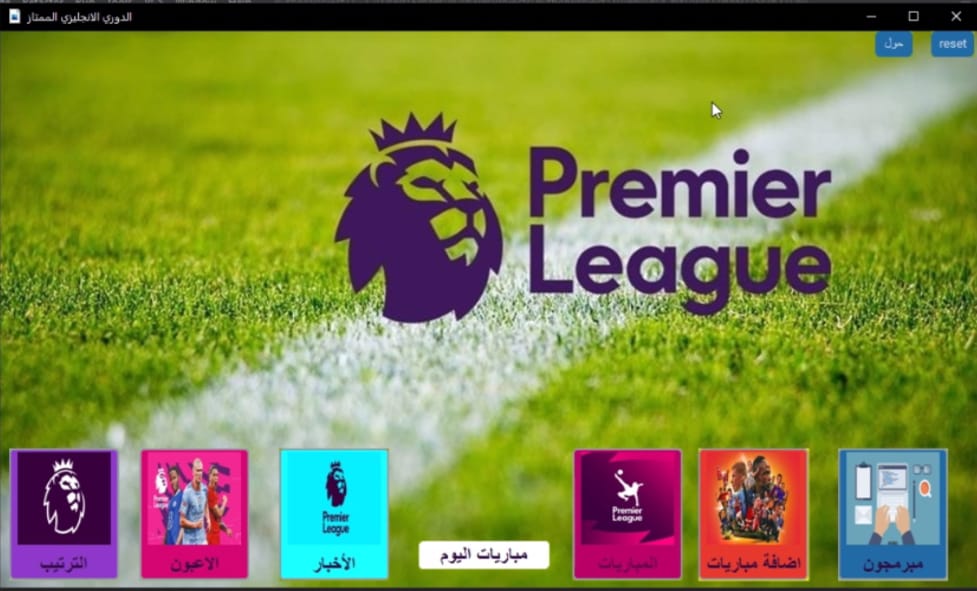


photo2 = ImageTk.PhotoImage(Image.open(r"k12.jpeg").resize((100,100)))

In the main window, the image taken from the file is defined and linked to a name and its dimensions are set.

bt = customtkinter.CTkButton(var, image=photo2, text='المباريات', text\_font=('tajawal', 16, 'bold'),command=lambda :matches(var), compound=TOP, border\_width=2, fg\_color='#b90066', width=118, bg\_color="#84a202",height=143,text\_color="#600048")

Then the desired button is defined and placed on the window and the text to be placed "Matches" is written and the font type "text\_font = ('tajawal', 16, 'bold')" is selected and then "image=photo2" is placed so that the image is placed on the button.

bt.place(x=615, y=450)

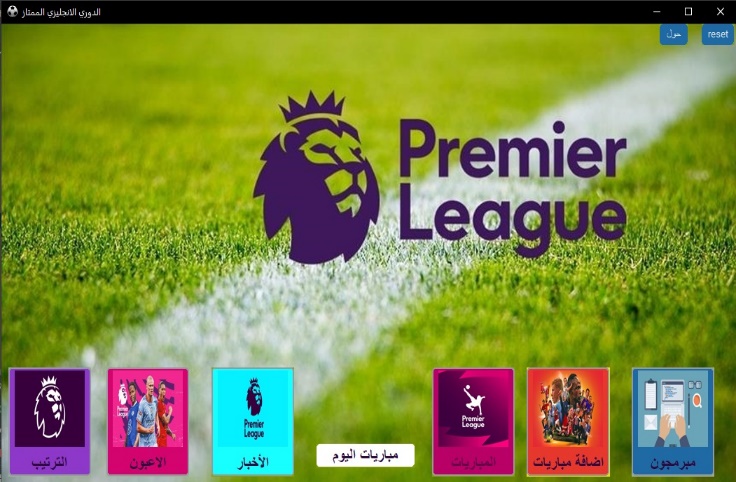
Then the button is located on the window and the initial value is selected from the northwest of the page and this process is repeated with the rest of the buttons.

And the "command" is used and then equals the name "class" and that when the button is pressed the user is moved to the page to be entered for example in this command "command=lambda:matches(var)" the user will be moved to the

"المباريات" window.

➢Arrangement window.

When you press the "الترتيب" button, a new frame is opened and the display system of the window is in the form of ordered and equal tables and each column in this table contains ("team name", "common name", "matches", "points").



Class Table:

# Initialize a constructor

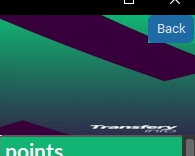
def \_\_init\_\_(self, gui,lis):

As shown in this picture, "class" is defined as a table and a function is made to take (self, gui, lis) and then Tim places both ("team name", 'common name', "matches", "points") inside "the list".

For i in range (20):  
 for j in range (4):  
 print (i)  
 w = 0

And "for loop" is made to make the table designed to contain (20 rows and 4 columns) in an organized and equal way.

And the order system of this initial table is alphabetical and then the order varies according to the number of points of the largest team These points that the team gets through the added matches window and based on which the points are calculated and if the number of points of two teams is equal to each other it takes the ranking according to the number of goals most for the team.

b = customtkinter.CTkButton(f1, text="Back", width=20,command=main2)

b.place(x=750,y=0)

var.update()

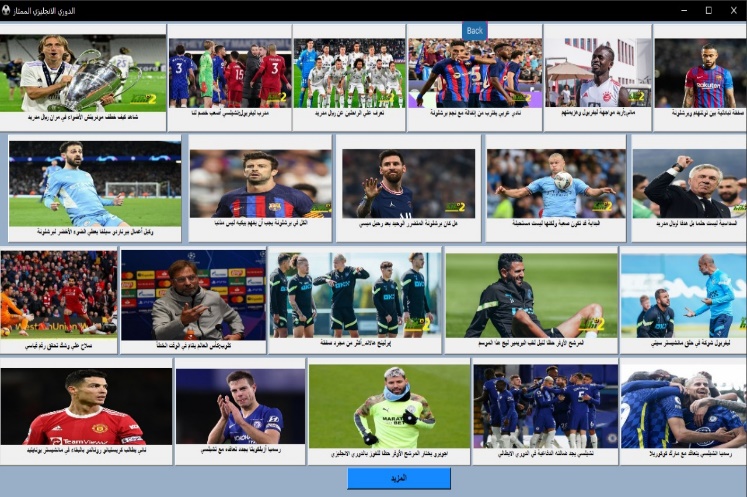
And the "back" button when

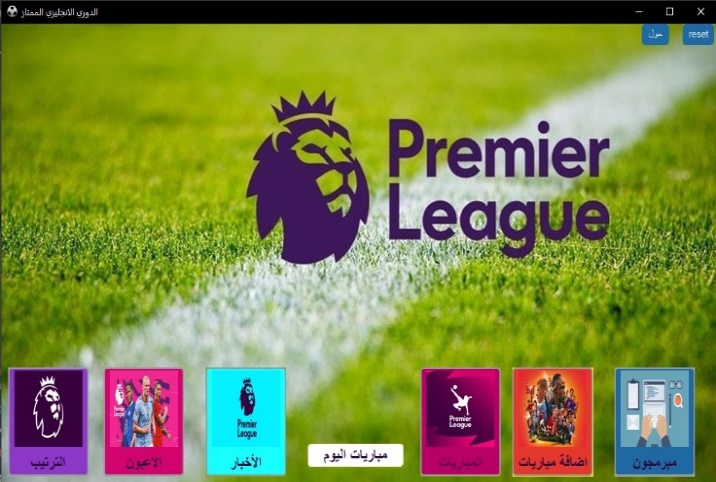
Pressed removes the "Order" and returns to the main window again without deleting any data in the league table and it is placed in the following dimensions(x=750,y=0).

The mechanism of action of the function is to make a button and it is called "back" and after selecting its place on the window is made "up data" so that when you press the button it returns to the main background.

➢News window.

When you press the "الاخبار" button, the main window is closed and a new frame is opened containing all the news of football, which is the news of "the transfer of players or the contract of a club with a new player or press reports on the reasons for the transfers or a player's statement.....etc".





Inside the news window there is football news worldwide each news is placed inside a button and under it there is a label and inside it there is the news and when you click on it opens a web browser page and shows you the news in exact detail.

res1 = photo1.subsample (5, 5)  
bt1 = Button(pro, image=res1, text="ماني:أريد مواجهة ليفربول وهزيمتهم", font=('tajawal', 8, 'bold'),  
 compound=TOP, width=190, height=150, command=ope2)  
bt1.place(x=1000, y=4)

The dimensions of each image are determined "res1 = photo1.subsample(5, 5)" and the desired image is selected and placed on the button "image=res1" and then the typed text is placed under each image to form all the news " text="ماني:أريد مواجهة ليفربول وهزيمتهم" " and then specifies the type and size of the font used "font=('tajawal', 8, 'bold')" and then places the dimensions of each button "compound=TOP, width=240, height=150" and then the button is linked to the link of the web page which when the user clicks on it takes it to the news page to be read the news in detail "command=ope3" in the end, the dimensions of the button on the window are specified "bt2.place(x=750, y=4)".

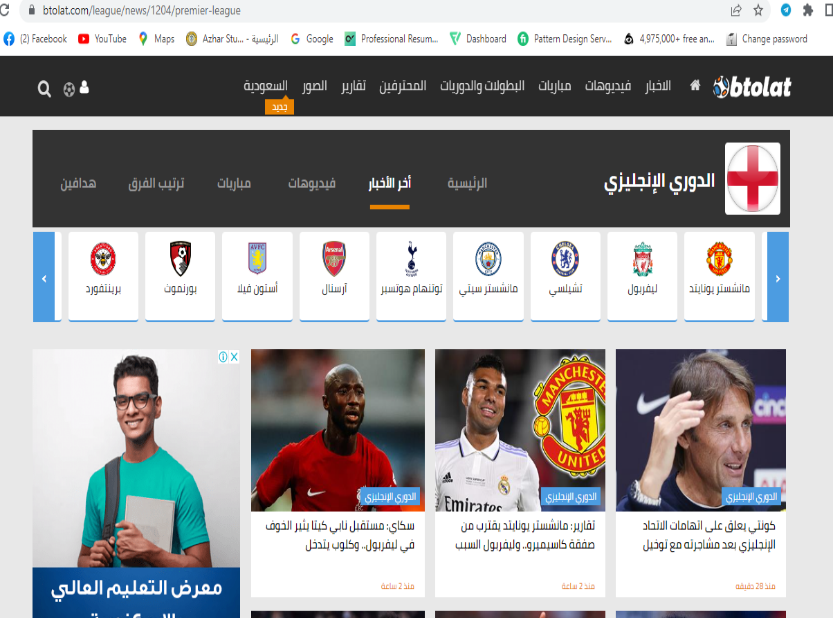
And this process is repeated with all the news that is returned to be downloaded by the window.

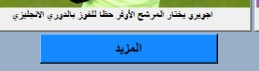
def ope1():  
 webbrowser.open('https://hihi2.com/2022/08/05/p1981265.html')  
  
def ope2():  
 webbrowser.open('https://hihi2.com/2022/08/05/p1981259.html')

The button is linked to the Internet browser by making a function and putting a name for it for example "open" and then “web browser .open” and then open a bracket and the link to the web page is written and then close the bracket and when the button is named and put "Command = name of the function of the page link".

This function is repeated with all the news that is returned to be downloaded by the window.

At the end of the window there is a button called "المزيد" when you click on this button is automatically moved to the page web browser and inside this page displays the daily news of the English Premier League and this page is updated day by day from the original site to display it.





C:\Users\user\Desktop\Capture.PNG7.PNGAnd the mechanism of action of this button is like the mechanism of working to a button in the window.

The "back" button when you press

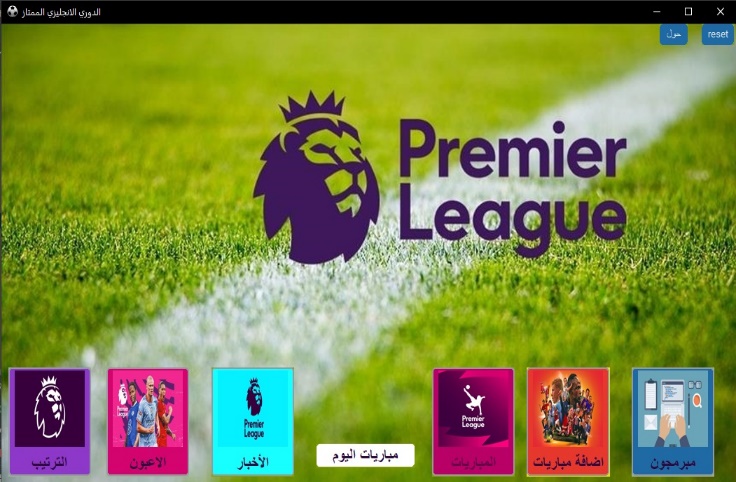
This button is exited the “الاخبار" window

And then returned to the main window.

The mechanism of action of the button is like the mechanism of action of the "Back" button in the order window.

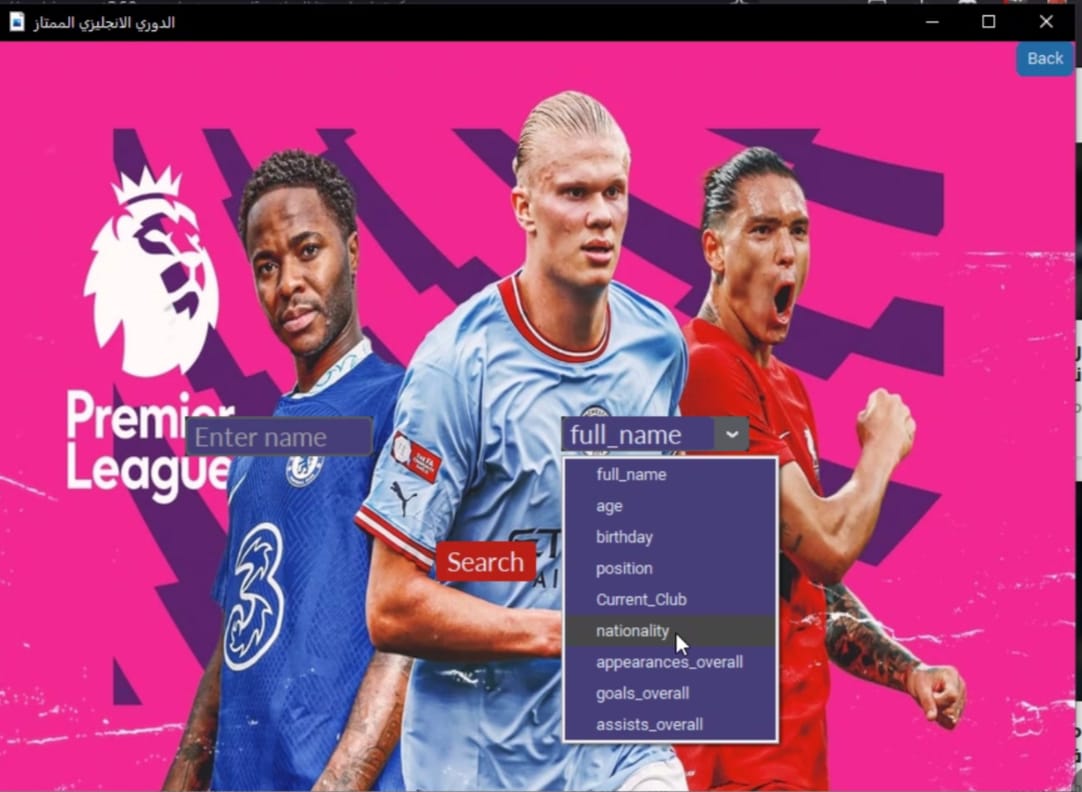
➢Players window.

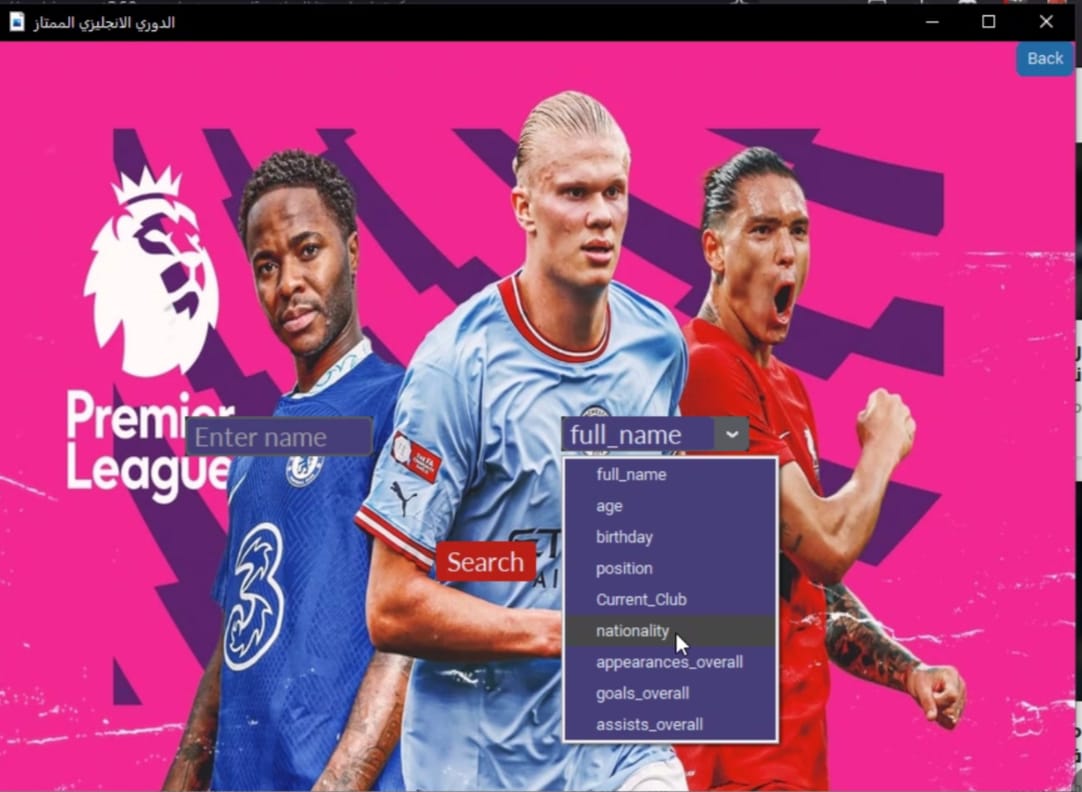
Inside this window the user can find out any information about the "players" within the English league through which you can search for "the age of the players, the nationality of the players, the players' club, the number of goals of the player or the number of goal making....... etc".



When entering the "Players" window, the user finds a "Search" button and two "Combo Box" through which the information that the user wants to inquire about is selected.

When you click on it, a menu appears inside it, and the object you want to inquire about (values=["full\_name","age","birthday","position","Current\_Club","nationality","appearances\_overall").



The combo box is defined as "c1 = custom tkinter. The CTk Combo Box size is then determined and the values are entered"["full\_name","Age","Christmas","Position","Current\_Club","Nationality","appearances\_overall","goals\_overall","assists\_overall" and then the font type "text\_ font=('lato', 16, '')"

Is selected and then we put instead of

The comb box "c1.place(x=450,y=300)".

CTk Entry is defined as

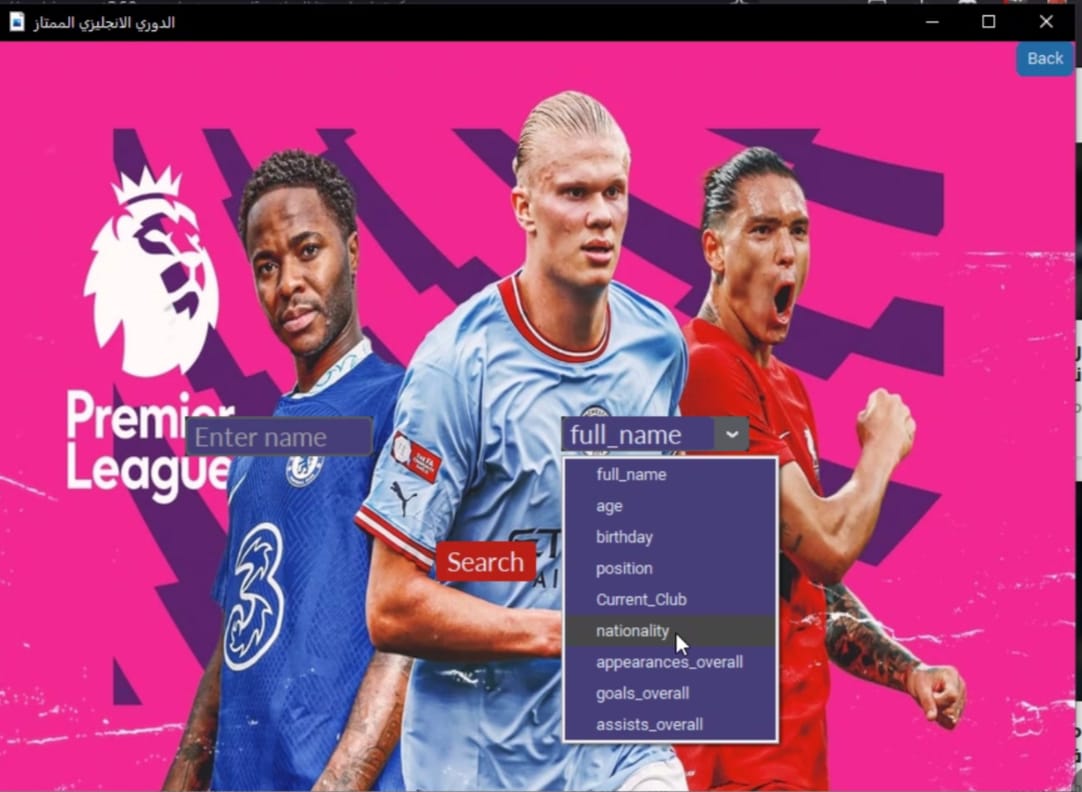
"c1=custom tkinter. Then the CTk Entry

Size is specified and the values

"Placeholder\_ text='Enter Name'" are entered and in this way it will become the word "Enter Name'" is transparent and when pressed disappears and the object to be typed by the user is written and then the font type "text\_ font= ('Lato', 16, '')" is selected and then we put instead of the CTk Entry "e1.place(x=150, y=300)".

The mechanism of action of the "Search" button is defined "b2=customtkinter. CTk Button" then the size of the button is selected and then the text is placed on the button "text="Search" and then the type and font size "text\_ font=('Lato', 16, '')" are selected then locate the button on the window "b2.place(x=350,y=400)".

When selecting the object to be

Searched for and the user is pressed on

The "Search" button, all the information

About the English league is displayed

Such as (names of players/teams/.....etc)

On a new window in the form of a table.

How it works when selecting the inputs

Teams=list(cr. execute(f" SELECT \* from players WHERE {choose} LIKE '%{text}%'"))

In "CTk Combo Box" and when you click on "Search"

It searches inside the "data base" and makes

Fitsh and then shows me

The desired results as a table.

"Tree view" is created so that

We can show the search

Results within the frame in

The form of organized and

Equal tables.

Style. configure("Tree view",  
 background="#c9c4cd",  
 foreground="black",  
 row height=55,  
 field background="#463e78")

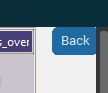
The background, color, and row and column size are all specified within the function.

Tree = ttk. Tree view(f3.interior, style="Treeview",column=("full\_name","age","birthday","position","Current\_Club","nationality","appearances\_overall","goals\_overall","assists\_overall"), show='headings', height=50)

Then the texts that will be written on each column are defined (("full \_ name ","age ","birthday" ,"position"……… etc).  
Then each text and its equal is defined by its own column.

|  |  |
| --- | --- |
| Tree .column("# 1", anchor=CENTER)  tree .heading("# 1", text="Name") | The first column is for the players' name. |
| Tree .column("# 2", anchor=CENTER ,width=70)  Tree .heading("# 2", text="age") | The second column is for the age of the players. |
| Tree .column("# 3", anchor=CENTER, width=70)  Tree .heading("# 3", text="birthday") | The third column is for the players' birthday. |
| Tree .column("# 4", anchor=CENTER, width=70)  tree .heading("# 4", text="position") | The fourth column is for the positions of the players on field. |
| Tree .column("# 5", anchor=CENTER, width=90)  tree .heading("# 5", text="Current \_Club") | The fifth column is for the players club. |
| Tree .column("# 6", anchor=CENTER, width=70)  tree .heading("# 6", text="nationality") | The sixth column is for the nationality of the players. |
| Tree .column("# 7", anchor=CENTER, width=70)  tree .heading("# 7", text="appearances \_overall") | The seventh column is for the number of matches the player has played. |
| Tree .column("# 8", anchor=CENTER, width=70)  tree .heading("# 8", text="goals \_overall") | The eighth column is for the number of player goals. |
| Tree .column("# 9", anchor=CENTER ,width=70)  Tree .heading("# 9", text="assists \_overall") | The ninth column is for the number of the player's assists. |

When you click on it, you will return to the "Players" window.



b = customtkinter. Ctk Button

(f3.interior, text="Back", width=20, command=page \_player,

Bg \_color="white")  
B. place(x=800,y=0)

C:\Users\user\Desktop\Capture.PNG11.PNGHow the function works after specifying the size of the button "width=20" and specifying the text inside it "back", the command "command=page \_player" is specified and this command is to return to the previous page "players" and then the font type "bg\_ color="white" is selected Then it locates the button " b. place(x=800, y=0)".

The mechanism of action of this button is similar to

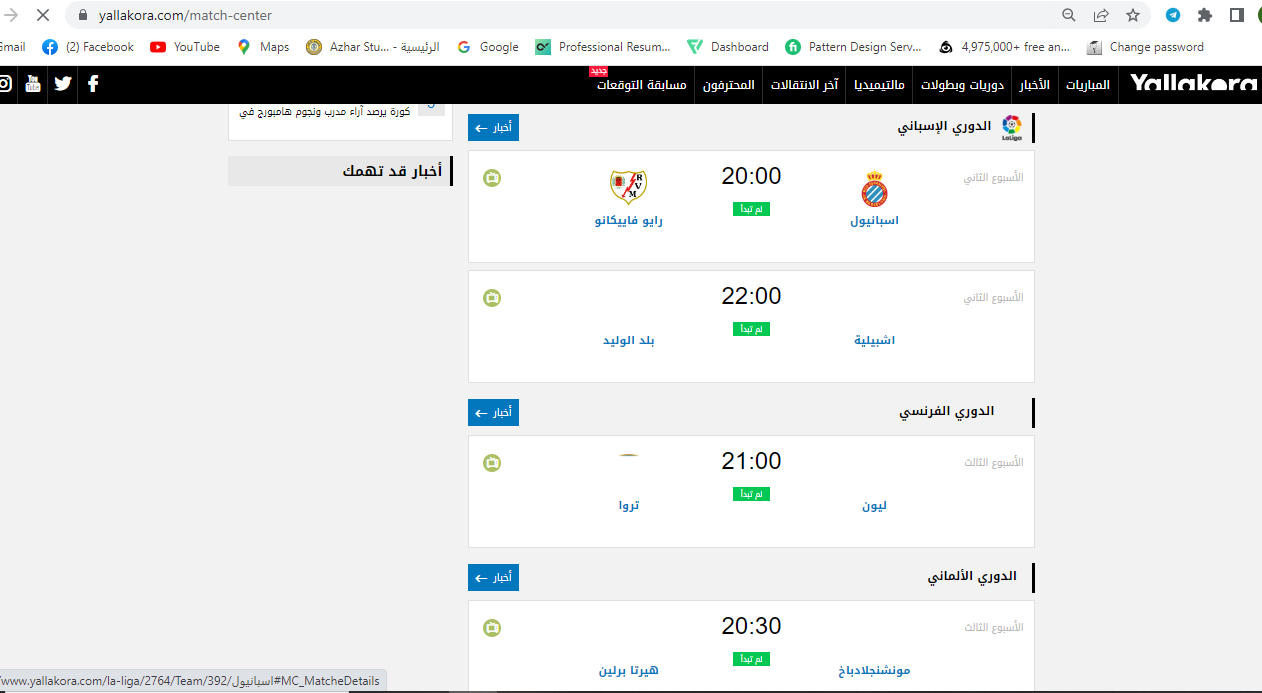
The mechanism of action of the button that was on

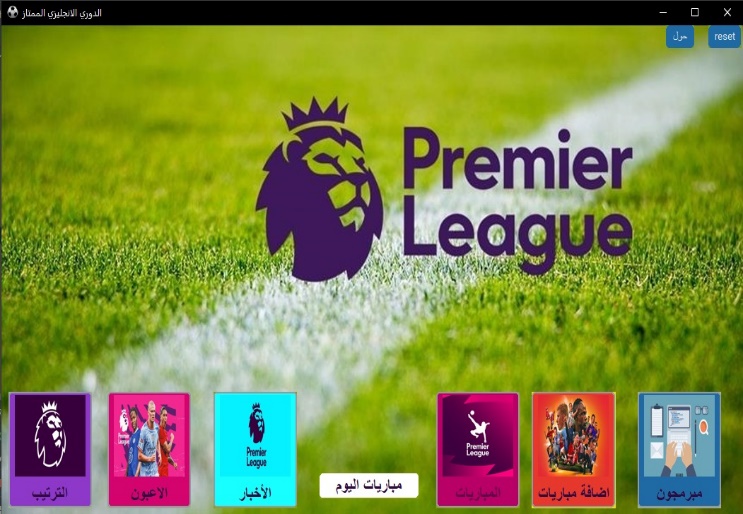
The previous page with a difference that when the user

Uses this button, he returns to the main window.

➢Today's Matches window.

When the user presses this button he is moved to a web page with all the daily matches of all football leagues.





bt4 = customtkinter CTkButton(var, text='مباريات اليوم', text \_color="#2f0c40", text\_ font=('tajawal', 16, 'bold'),border\_ width=0, fg \_color='white', command=open1,cursor='mouse',bg\_ color="#84a202")

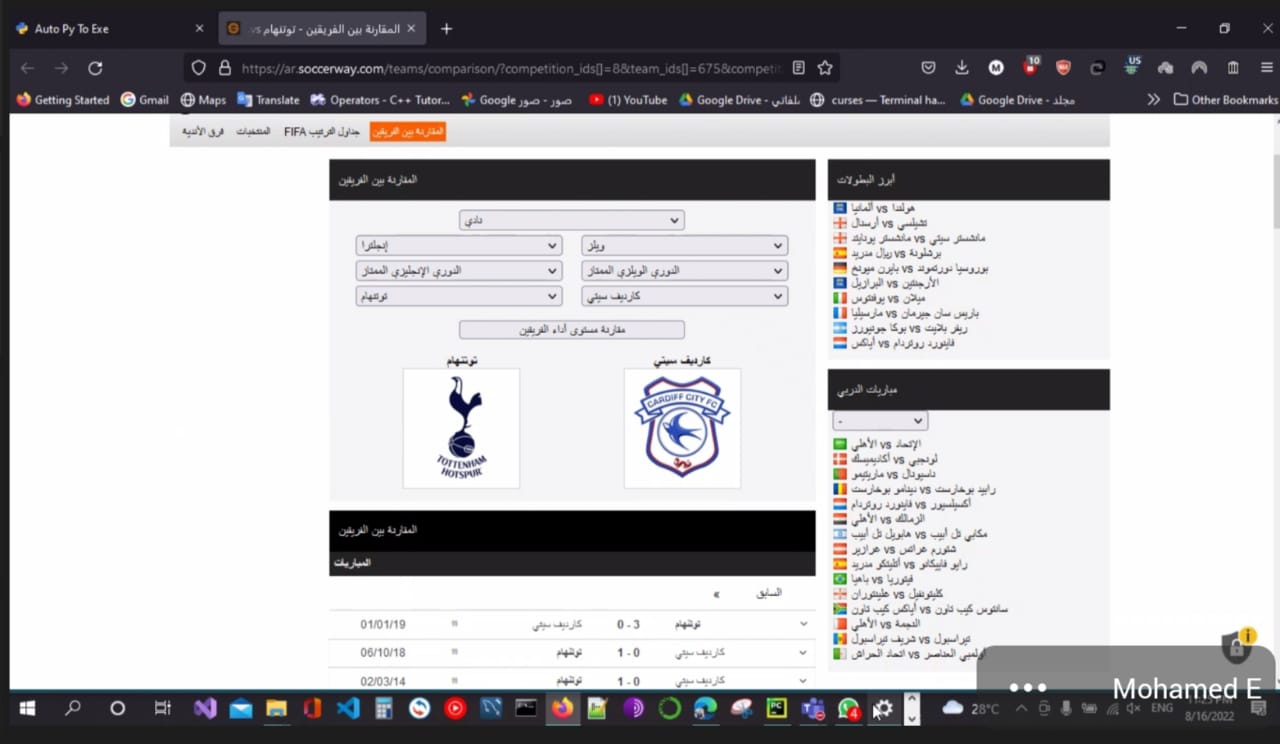
bt4.place(x=450, y=550)

Tim Action button calls "Matches of the Day" then the size and type of font used to type "bg \_color="#efde05", text \_font=('tajawal', 16, 'bold')" is selected and then a command "command=open1" is placed which linked the button to the web page and then the button is positioned on the main window.

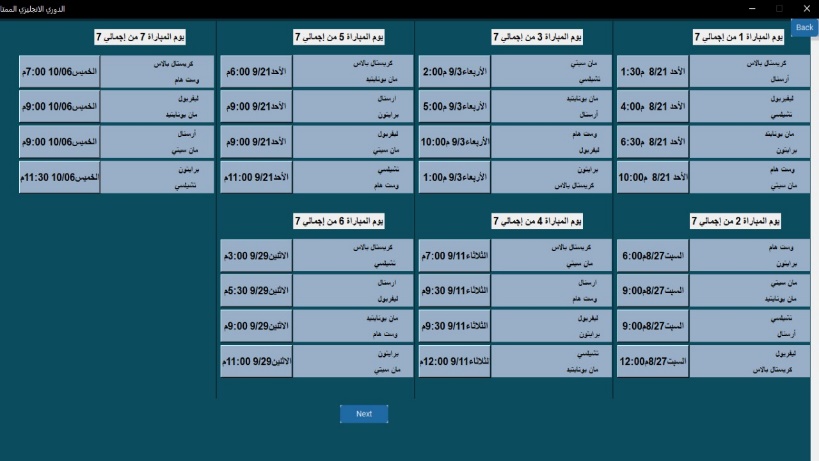
➢ Match window.

When you click on the "Matches" button, all the timing of the Premier League matches is displayed.



Another feature is that when the user clicks on a specific match date between two teams, he goes to a web page with the date of the two teams' confrontation and has win-loss statistics and draws each team over the other.

The window mechanism of action "Games" button "bt =customtkinter" is designed. CTk Button" is then selected the image and text placed on the button "image=photo2, text='matches'" and then the font type used " text\_ font=('tajawal', 16, 'bold')" is chosen and then the command "command=lambda: matches( var ), compound=TOP" is placed so that when the user presses the button it is moved to the "Matches" window and then the dimensions of the button "fg\_ color='#b90066', width=118, bg\_ color="#84a202",height=143" and then its place is selected on the main window "bt. place(x=615, y=450)".

def open1():  
 web browser .open(  
 'https://www.alremontada.com/h2h/%D8%A7%D8%B1%D8%B3%D9%86%D8%A7%D9%84-%D9%88%D9%83%D8%B1%D9%8A%D8%B3%D8%AA%D8%A7%D9%84-%D8%A8%D8%A7%D9%84%D8%A7%D8%B3?fixture\_id=157105')  
def open2():

First of all, functions will be defined

And each function will be identified

By a link to the web page, which will

Later be linked to the button that

The user will click on and then these

Functions will all be placed inside the "class".

l1 = Label(f1, text='يوم المباراة 1 من إجمالي 7', font=('tajawal', 12, 'bold'))

l1.place(x=130, y=10)

After defining web pages within "class", "Label" is defined to be written on the day on which the match will be held, then the text and the type of font used in the writing are written "text='match day 1 out of a total of 7', font=('tajawal', 12, 'bold')" and then the text is located on the window "l1.place(x=130, y=10)".

Then the name of the two teams that

Will face each other is placed inside

The "Label" and then the names of

The teams are specified and written

"Text='Crystal Palace' and text='Arsenal'"

And the type and font size "font=('tajawal', 10, 'bold')" are selected and then located on the window "l2.place(x=90, y=0) and l3.place(x=130, y=25).

After determining the teams, the

"bt1=Button" button is made on which

The match date is written

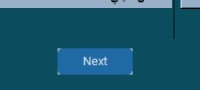
"Text='Sunday 8/21 M1:30'" and then

The type and font size used

"Font=('tajawal', 12, 'bold')" is selected and then the command "command=open1" is placed and this command is placed when the user presses the button will show him the date of the confrontations of the two teams.

And these functions are applied to the rest of the matches.

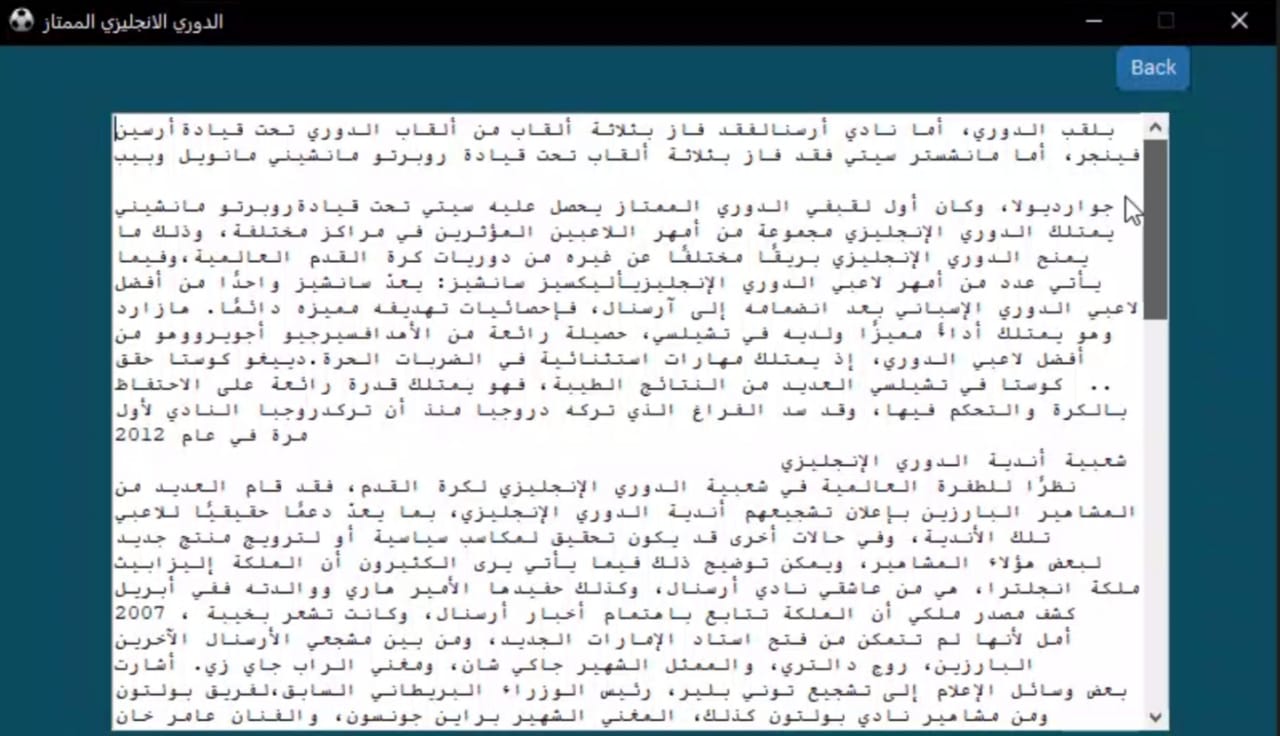
The "Next" button is used to display the rest of the matches between each team.

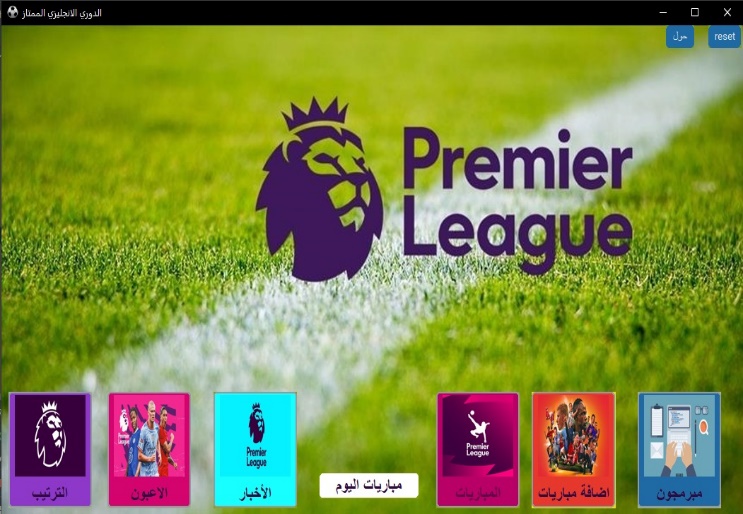
"b2=customtkinter". CTk Button(root, text="Next", width=80, command=lambda : core(root), bg\_ color="#98AFC7")

b2.place(x=600, y=600)"

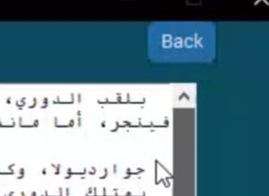
➢About window

Inside this window there is general information about the English Premier League.





The "customtkinter" button is defined. CTk Button" is then written as the text "), text="about"," and the type and font size of the user are specified "width=40, bg\_color="#84a202")" and then a command is placed "var, command=lambda: info (var), "to the user to go directly to the information page The button is located on the window "b7.place(x=940, y=0)".

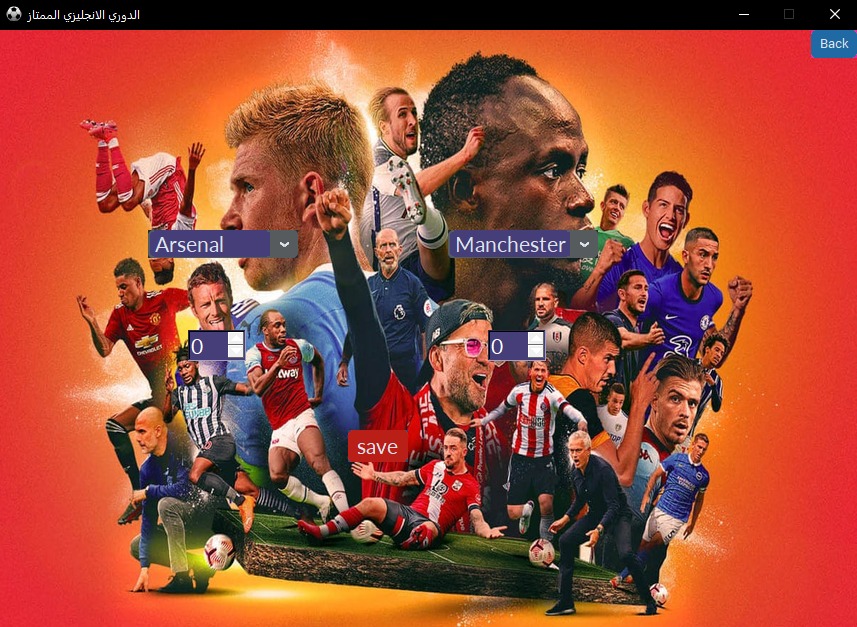
Then "laern = """"" is defined and then placed ""in order to write the text to be written"""

The "Back" button is used to return to

The main menu.

➢Add matches window.

This window only allows project supervisors who can control the results of matches to put the results and on this approach the correct arrangement of the league is made.

Inside this window is defined the number of 2 comb box "C1=customtkinter. CTkComboBox" through which the two teams facing each other "variable=s1,values=l1" are selected and then the command "command=lambda s:k1(s1.get(),c2)" is specified through which the two teams will be selected and their data will be saved for the results to be compared later, and then their respective locations will be located g= on the window"

c1.place(x=450,y=200)" and

After the teams are determined

, the results of each difference

Are determined by"sp1=Spin box"

Which will save the result, so

The number of 2 "sp1=Spin box"

Is defined, then their size is

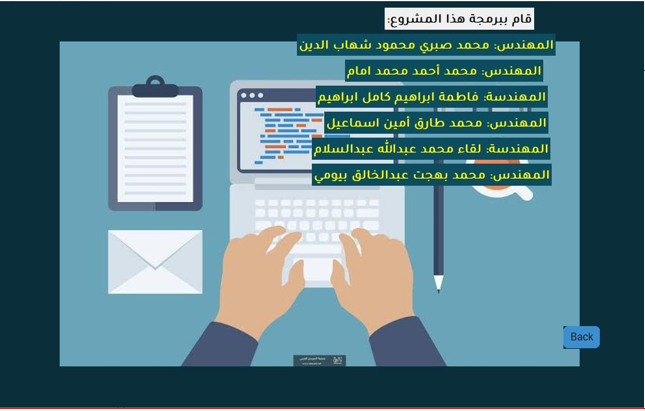
Determined and then linked to the team written in the comb box directly above it and is linked through this command "Print(list(map(lambda x: x[0],cr .execute("select common \_name from Teams2").fetchall())))".

The results are then compared together if the first team results greater than the second team "if(res1>res2):" It works to increase the number of the first team by an average of 3 points "cr. execute(f "update Teams2 set points=points+3 where common \_name='{team1}'")" and if the second team has more goals than the number of goals of the first team "elif(res2>res1):" The number of points in the league table of the second team is increased "cr. execute(f "update Teams2 set points=points+3 where common\_ name='{team2}'")

If the number of goals of each team equals the other, one point will be calculated for each team.

➢Programmers window.

This window shows the names of the makers of this program.



➢Reset window.

Through this button, the league is completely reset and all the values inside the ranking table are zeroed.

And the mechanism of action of this function is to zero points In the table, the values of match results are zeroed "cr. execute ("update Teams2 set matches\_ played=0, points=0")"

