Indian Institute of Information Technology, Nagpur



Attendance Record and Management

ITW-1 project

Group Members- Under the guidance of-

RAJ ARYAN [BT19CSE043] Mayuri Digalwar

PRATEEK SHENDE [BT19CSE051]

SARTHAL GUPTA [BT19CSE054]

HEMANSHU CHAUDHARI [BT19CSE056]

KRISH RUTAGI [BT19CSE089]

Acknowledgement

Table of contents

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Building the graphical Interface

-Bitmapicon

iconbitmap(bitmap) sets the icon of the window/frame widget to bitmap. The bitmap must be an ico type, but not png or jpg type, otherwise, the image will not display as the icon.

-Geometry

The geometry method is a fundamental one which decides the size, position and some other attributes of the screen layout we are going to create.

-canvas

The Canvas is a rectangular area intended for drawing pictures or other complex layouts. You can place graphics, text, widgets or frames on a Canvas.

-Scrollbar

Canvas is not updated automatically when its content is modified, so we need to define it and update it manually using the scrollregion argument:

Almost all these widgets will work like normal Tkinter widgets (i.e. you place them in their container and then use Pack or Grid). However, the scrollable frame will behave differently. We will create it, giving it the canvas as its container, but instead of using Pack or Grid, we will using the canvas' create\_window method to create a window within the canvas that shows the scrollable frame's content.

he values (0, 0) tell the canvas on which position to draw the window. The argument anchor="nw" tells the canvas to place the frame's top left corner on position (0, 0).

Finally, we have to configure the canvas so that when its y-position changes, the scrollbar moves:

That means that we'll be able to set the Canvas size to whatever we want, and then when we scroll we'll be moving along the window inside it.

In order to do so, we must also specify what the scrollable area will be. Often, the scrollable area matches the contents of the inner window—but it can also be different if we wish.

-Frame label

A labelframe is a simple container widget. Its primary purpose is to act as a spacer or container for complex window layouts.

This widget has the features of a frame plus the ability to display a label.

-Buttons

The Button widget is used to add buttons in a Python application. These buttons can display text that convey the purpose of the buttons. You can attach a function or a method to a button which is called automatically when you click the button.

-Dropdown

Dropdowns are toggleable, contextual overlays for displaying lists of links and more.

The value selected from the options provide in the dropdown menu can be access from the variable assigned to it by using the method, variable.get().

-Input box

The entry widget is used to enter text strings. This widget allows the user to enter one line of text, in a single font. The value entered in the input box can be access from the variable assigned to it by using the method, variable.get().

-checkbox

The Checkbutton widget is used to display a number of options to a user as toggle buttons. The user can then select one or more options by clicking the button corresponding to each option.

The value selected from the options provide in the dropdown menu can be access from the variable assigned to it by using the method, variable.get().

-withdraw

Removes the window from the screen, without destroying it.

-deiconify

Displays the window, after using either the iconify or the withdraw methods.

-Using lambda to call multiple functions at a time

-frame.bind

This is about binding events to functions and methods so that when the event occurs that specific function is executed.

Reading From the excel record

-what is dataframe

-how we read

-using used a list of dataframe

-dataframe.parse

-index col = none

Displaying the data from dataframe to UI

-dynamically selecting with list item to show

-looking for empty columns to take input for that day

-checkbox variable

-dropdown variable

- variable.get()

-different rows

-checkboxes along with name of student

-

Calculating %

-selecting the list item based on the selection of course

-reading number of presents in a single row

-reading total classes attended

-calculating

- displaying the output with name of student

-

Updating dataframe list

-taking input from the checkbox

-present variable

-absent variable

-markpresent()

-markabsent()

Saving the record

-save()

-excelwriter

-why all list object being written to the file

Modifying the record

* Add student
* Taking input from the input box
* Updating dataframe
* Save
* Delete student
* Taking input roll number
* Taking name input
* Searching the record
* Deleting the row
* Save()

References