LOVELY PROFESSIONAL UNIVERSITY

Create registration and loginform using python

PYTHON PROGRAMMING (INT 213)

Name : Akash Soni

Registration Number :12102161

Program : CSE B.Tech

Semester: Third

School: School of Computer Science

andEngineering
Name of the University:
LovelyProfessional University
Date of submission:15th NOVEMBER
2021
Professor Name: Prof. Md. Imran
Hussain

Lovely Professional University
Jalandhar, Punjab, India.

Create Registration and Login form using Python

ABSTRACT:-

Gone are the days of long registration queues and paper forms to sign up for an endurance event. An increasing number of event directors from both largeand small events are embracing online registration services as an added-value for both them and their participants. Online registration not only improves efficiency and eliminates unnecessary paperwork, it also maximises participation and improves marketing capabilities while allowing participants to sign up when and where it is most convenient for them from any Internet-enabled computer.

ACKNOWLEDGEMENT:-

I would like to thank my mentor - Prof. Md.

Imran Hussain for his advice and inputs on this project. Many thanks to my friends and seniors as well, whospent countless hours to listen and provide feedbacks

Table of Contents

INTRODUCTION:-

1.1 Context

This project has been done as part of my course forthe CSE(H) at Lovely Professional University .

Supervised by Sagar Pande, I have three months tofulfill the requirements in order to succeed the module.

1.2 Motivations

Being extremely interested in everything having a relation with the Python, the group project was a great occasion to give us the time to learn and confirm our interest for this field. Designing an effective and engaging graphical user interface (GUI) that will help your digital presence to shine in today's competitive online marketplaces is a much more involved process than you might initially imagine. That's why I decided to conduct my projectaround Tkinter.

1.3 Idea:- Modern society heavily depends on the abilities of computers, information processing, and Information Technology. As such, since access to information occurs mainly through digital means andmedia, the way information is arranged and presented is crucial. Because this need for fast access and easy arrangement arose, software companies started to work on various graphical userinterfaces (or GUI for short).So I chose to make Registration and Login Form as an approach. The goal was that the very best user interfaces enhancethe relationship between brand and audience and importantly, this value can exist in a variety of forms.

LIBRARIES & Modules:-

Tkinter:- Tkinter is a Python binding to the Tk GUItoolkit. It is the standard Python interface to the TkGUI toolkit, and is Python's de facto standard GUI.Tkinter is included with standard GNU/Linux, Microsoft Windows and macOS installs of Python.The name Tkinter comes from the Tk interface.

Pillow:- Python Imaging Library (expansion of PIL) is the de facto image processing package for Pythonlanguage. It incorporates lightweight image processing tools that aids in editing, creating and saving images. Support for Python Imaging Library got discontinued in 2011, but a project named pillowforked the original PIL project and added Python 3.xsupport to it. Pillow was announced as a replacement for PIL for future usage. Pillow supportsa large number of image file formats including BMP, PNG, JPEG, and TIFF. The library encourages adding support for newer formats in the library by creating new file decoders.

RegEx:- It is a special sequence of characters thatuses a search pattern to find a string or set of strings. It can detect the presence or absence of a text by matching with a particular pattern, and alsocan split a pattern into one or more sub-patterns.

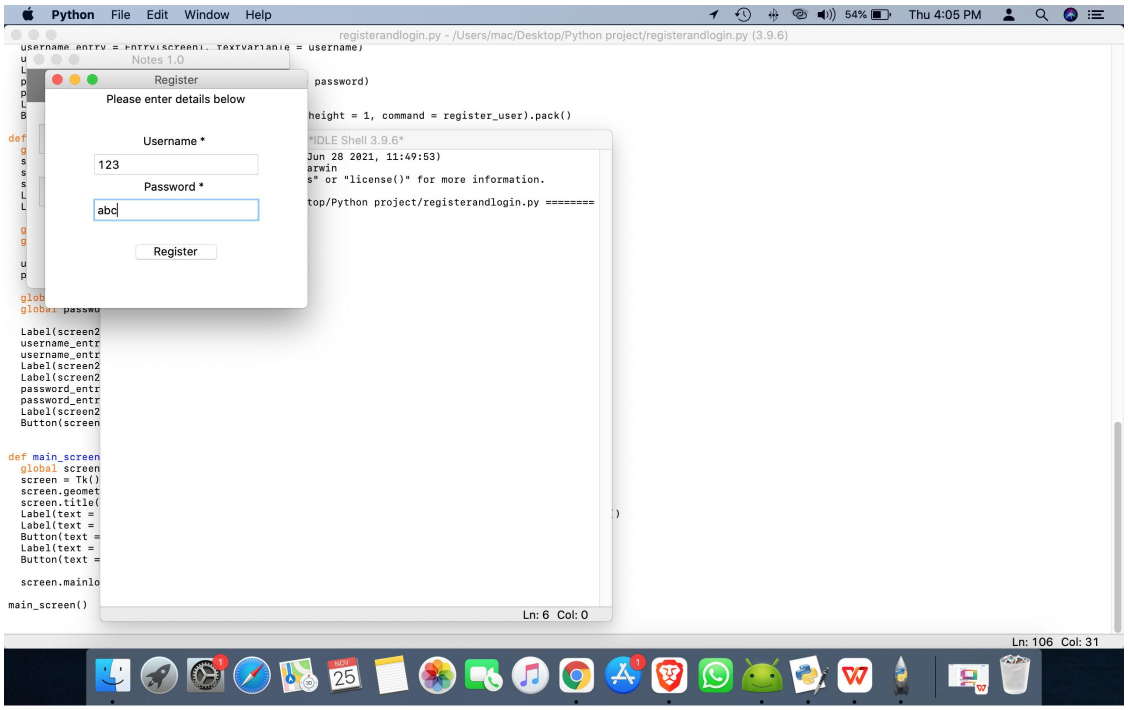
Python provides a re module that supports the useof regex in Python. Its primary function is to offer asearch, where it takes a regular expression and a string. Here, it either returns the first match or elsenone.

PyMySQL:- It is an interface for connecting to a MySQL database server from Python. It implementsthe Python Database API v2.0 and contains a
pure-Python MySQL client library. The goal of PyMySQL is to be a drop-in replacement for MySQLdb.

Messagebox:- It is a module in python which provides a different set of dialogues that are used todisplay message boxes, showing errors or warnings,widgets to select files or change colors which is a pop-up box with a relevant message being displayed.

SCREENSHOTS:-

1.Register Home page:-



Register

Please enter details below

Username *

123

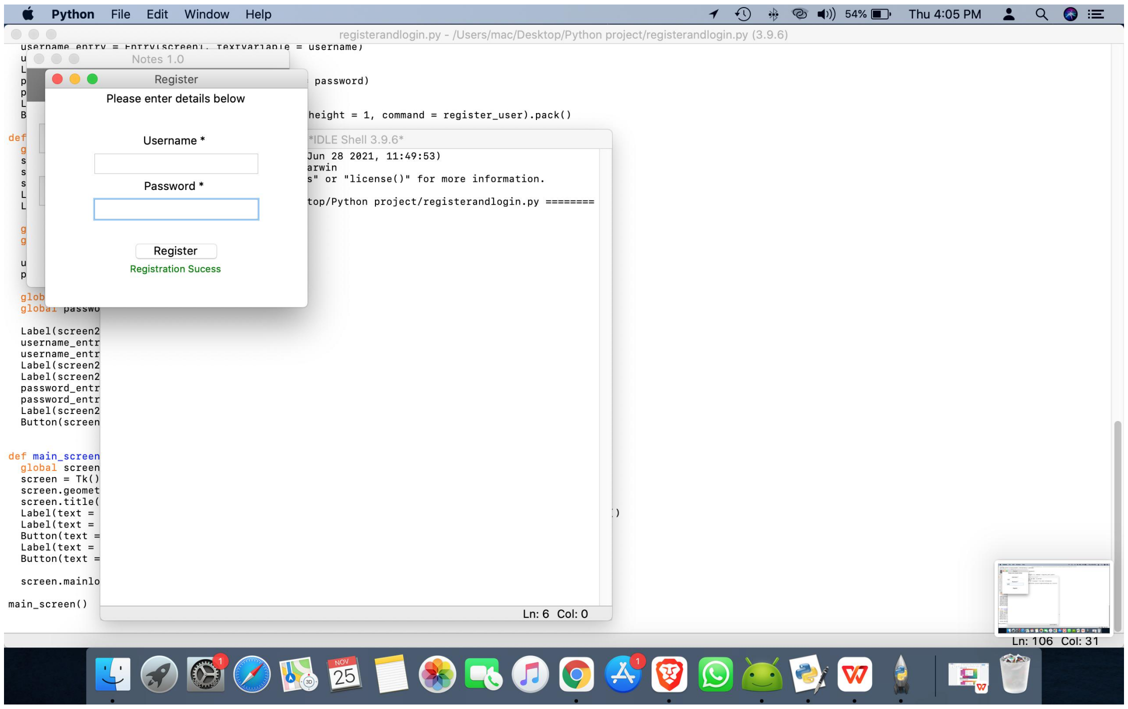
Password *

abc

```
username_entry = Entry(screen1, textvariable = username)
 password)
 height = 1, command = register_user).pack()

def main_screen
  global screen
  screen = Tk()
  screen.geomet
  screen.title(
  Label(text =
  Label(text =
  Button(text =
  Label(text =
  Button(text =

  screen.mainlo

main_screen()
```

```
*IDLE Shell 3.9.6*
Jun 28 2021, 11:49:53)
arwin
s" or "license()" for more information.

top/Python project/registerandlogin.py ========
```

2.Registration successful:-



Register

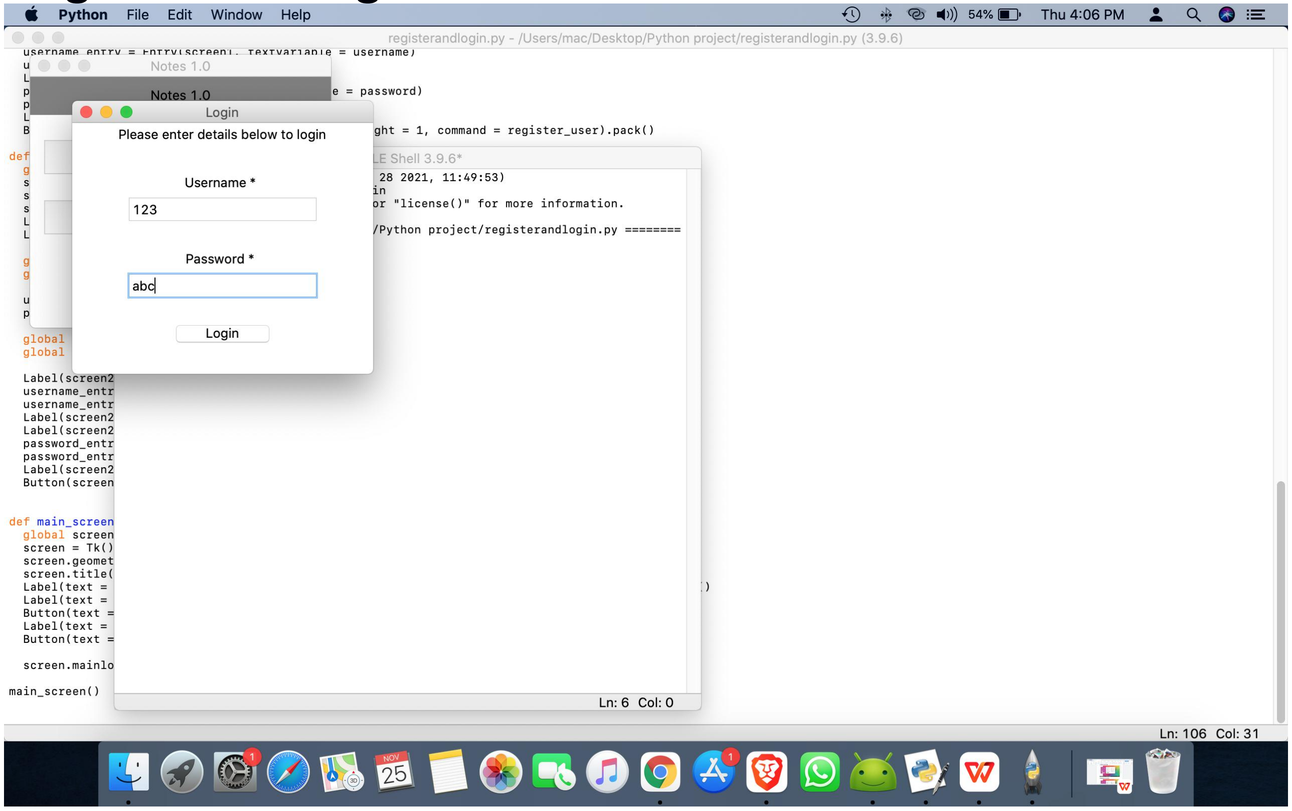
Please enter details below

Username *

Password *

Registration Sucess

3.Login Home Page:-



Please enter details below to login

Username *

123

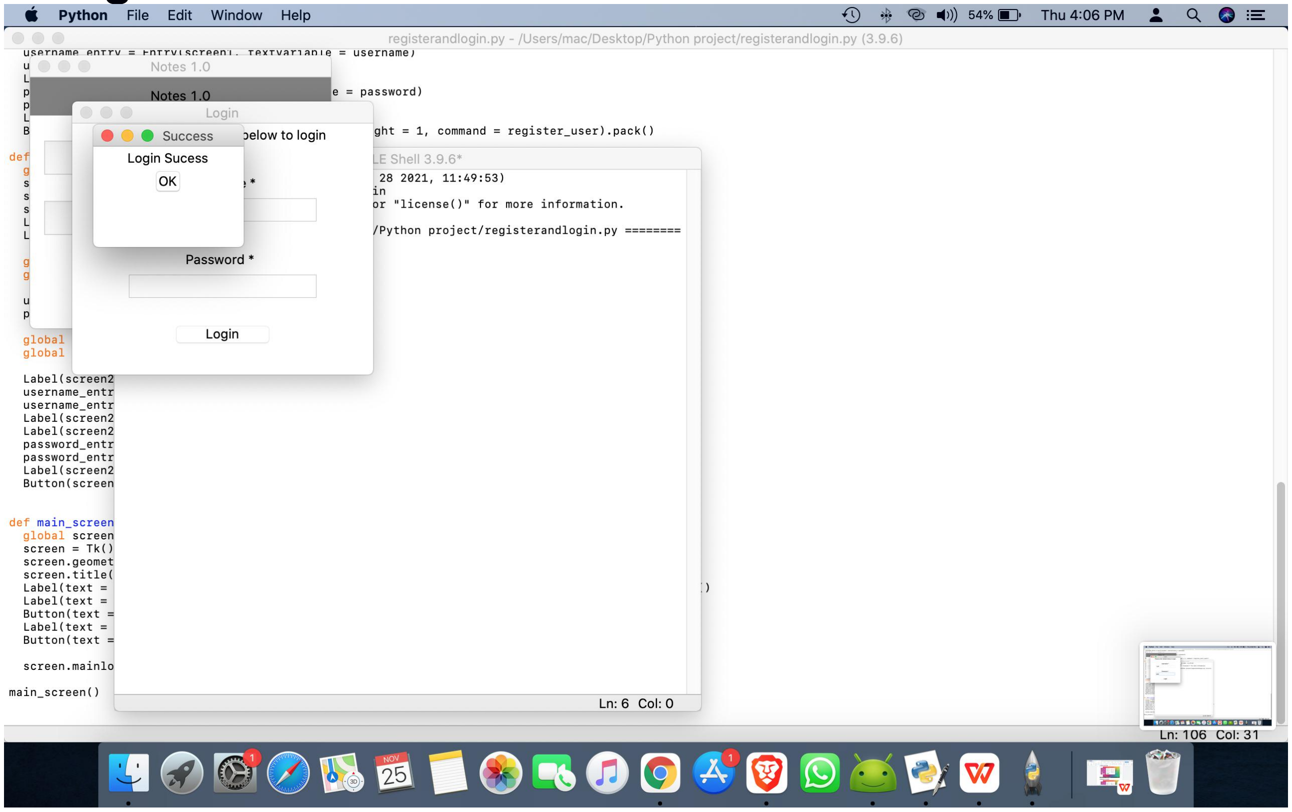
Password *

abc

Login

```
username_entry = Entry(screen1, textvariable = username)
                                       e = password)
                                       ght = 1, command = register_user).pack()
def
                                        28 2021, 11:49:53)
                                       in
                                       or "license()" for more information.
                                       /Python project/registerandlogin.py ========
  global
  global

  Label(screen2
  username_entr
  username_entr
  Label(screen2
  Label(screen2
  password_entr
  password_entr
  Label(screen2
  Button(screen

def main_screen
  global screen
  screen = Tk()
  screen.geomet
  screen.title(
  Label(text =
  Label(text =
  Button(text =
  Label(text =
  Button(text =

  screen.mainlo

main_screen()
```

4.Login Successful:-



Login Sucess

OK

below to login

Password *

Login

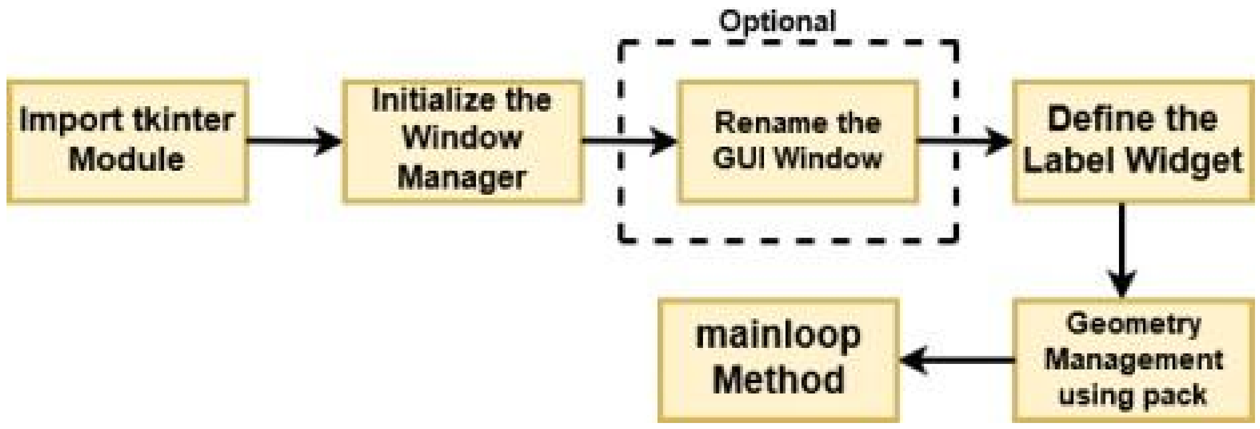
```
username_entry = Entry(screen1, textvariable = username)
                                       e = password)
                                       ght = 1, command = register_user).pack()
def
                                        28 2021, 11:49:53)
                                       in
                                       or "license()" for more information.
                                       /Python project/registerandlogin.py ========
  global
  global

  Label(screen2
  username_entr
  username_entr
  Label(screen2
  Label(screen2
  password_entr
  password_entr
  Label(screen2
  Button(screen

def main_screen
  global screen
  screen = Tk()
  screen.geomet
  screen.title(
  Label(text =
  Label(text =
  Button(text =
  Label(text =
  Button(text =

  screen.mainlo

main_screen()
```

Introduction to GUI With Tkinter in Python

- Tkinter commonly comes bundled with Python, using Tk and is Python's standard GUI framework. It is famous for its simplicity and graphical user interface. It is open-source and available under the Python License.

Note: `Tkinter` comes pre-installed with Python3, and you need not bother about installing it.

Now, let's build a very simple GUI with the help of Tkinter and understand it with the help of a flow diagram.



Flow Diagram for Rendering a Basic GUI

Let's break down the above flow diagram and understand what each component is handling!

- First, you import the key component, i.e., the `Tkinter` module.
- As a next step, you initialize the window manager with the `tkinter.Tk()` method and assign it to a variable. This method creates a blank window with close, maximize, andminimize buttons on the top as a usual GUI should have.
- Then as an optional step, you will `Rename` the title of thewindow as you like with `window.title(title_of_the_window)`.
- Next, you make use of a `widget` called `Label`, which is used to insert some text into the window.
- Then, you make use of Tkinter's `geometry management` attribute called `pack()` to display the widget in size itrequires.
- Finally, as the last step, you use the mainloop() method todisplay the window until you manually close it. It runs aninfinite loop in the backend.

Widgets

Widgets are similar in spirit to `elements` in HTML. You will find different types of widgets for different types of elements in the Tkinter. They are standard GUI elements and provide the user with controls like buttons, text, menus, and text boxes.

- Button: Button widget has a property for switching on/off. When a user clicks the button, an event is triggered in the Tkinter.

  Syntax: button_widget = tk.Button(widget, option=placeholder) where `widget` is the argument for the parent window/frame while `option` is a placeholder that can have various values like foreground & background color, font, command (for function call), image, height, and width of button.
- Canvas: Canvas is used to draw shapes in your GUI and supports various drawing methods.

  Syntax: canvas_widget = tk.Canvas(widget, option=placeholder) where `widget` is the parameter for the parent window/frame while `option` is a placeholder that can have various values like border-width, background color, height and width of widget.

- Checkbutton: Checkbutton records on-off or true-false state. It lets you can select more than one option at a time and even leave it unchecked.

  Syntax: checkbutton_widget = tk.CheckButton(widget, option=placeholder) where `widget` is the parameter for the parent window/frame while `option` is a placeholder that can have various values like title, text, background & foreground color while widget is under the cursor, font, image, etc.

- Entry: Entry widget is used to create input fields or to get input text from the user within the GUI.

  Syntax: entry_widget = tk.Entry(widget, option=placeholder) where `widget` is the parameter for

  the parent window/frame while `option` is a placeholder that can have various values like border-width, background color, width & height of button etc.

- Frame: Frame is used as containers in the Tkinter forgrouping and adequately organizing the widgets.

Syntax: frame_widget = tk.Frame(widget,

option=placeholder) where `widget` is the parameter for

theparent window/frame while `option` is a placeholder

that

can have various values like border-width, height & width of

widget, highlightcolor (color when widget has to

befocused).

- Label: Label is used to create a single line widgets like

  text,images, etc.

  Syntax: label_widget = tk.Label(widget,

  option=placeholder)where `widget` is the parameter for

  the parent window/frame while `option` is a placeholder

  that can

  have various values like the font of a button, background

  color, image, width, and height of button.

Geometry Management

All widgets in Tkinter have some geometry measurements.

Thesegeometry measurements allow you to organize the

widgets and throughout the parent frames or parent widget area.

For this purpose, Tkinter provides you with three main geometrymanager classes:

- pack(): It organizes the widgets in a block manner, and the complete available width is occupied by it. It's a conventional method to show the widgets in the window. ●

grid(): It organizes the widgets in a table-like structure. You will learn about it in detail later in this tutorial.

- place(): Its purpose is to place the widgets at a specific position as instructed by the user in the parent widget.

Organizing Layout And Widgets

In this section of the tutorial, you will make use of both **`geometry`**

and **`widgets`**, and let's see the magic of Tkinter.

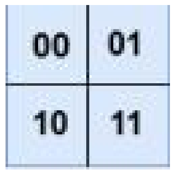
To arrange the layout in the **`window`**, you will use a Frame widget class. Let's create a simple program to see how the Frame works.

- You will define two frames - top and bottom with the help of the **`pack`** class. The Frame class will help in creating a division between the window. Basically, a single-window will be replicated twice as top and bottom in the form of a Frame.

- Finally, you will create four buttons in the window, two for each frame. You can name and color your buttons as you likeas a parameter.

Grid

Much like a `Frame`, a grid is another way to organize the widgets.It uses the Matrix row-column concept. Let's draw some analogybetween the `grid` class and the row-column idea with the help of the below diagram.



A Square Matrix (2x2)

Grid primarily takes two parameters `row` and `column`. As shown in the above figure, imagine `00` corresponds to the first buttonwhile `01` to the second. And to place two buttons side-by-side,`grid` will take row and column parameters as `00` and `01`, respectively.

Let's make use of `checkbutton` to understand how `grid` classworks. You will define two checkbuttons and specify some textfor both the buttons. The state of the check buttons will be decided by the `onvalue` and `offvalue` while the current stateof the checkbutton will be tracked by `IntVar()`, which will be

stored in a separate variable. When the `offvalue=1` and the `onvalue=0` corresponding checkbutton will be ticked.

Now coming to `grid` class, you will pass a parameter `row`, which will position the button in the first row if `row=0` and in the second row if `row=1`.

Binding or Command Functions

Binding or Command functions are those who are called whenever an event occurs or is triggered.

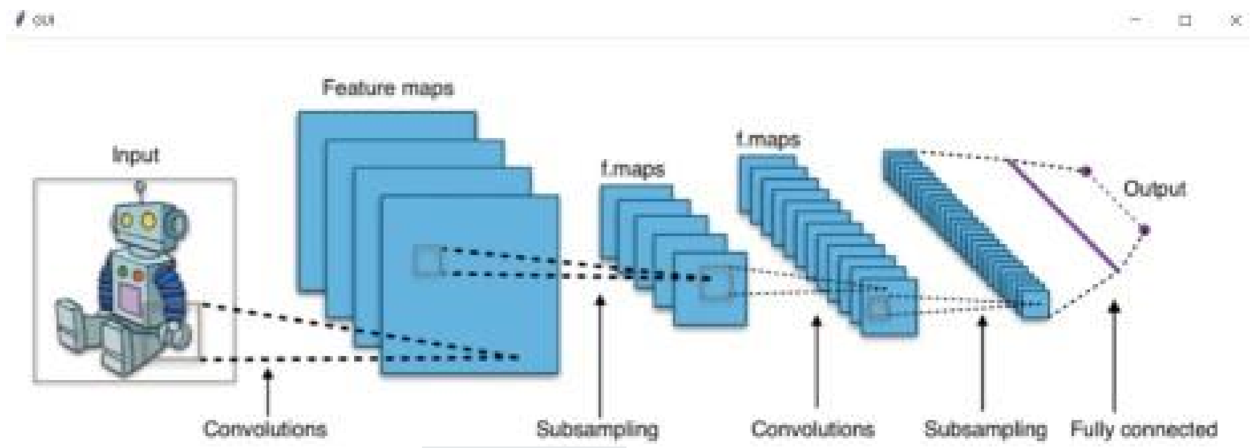
Alert Boxes

You can create alert boxes in the Tkinter using the `messagebox`method. You can also create questions using the `messasgebox`method.

Here you will create a simple alert-box and also create a question. For generating an alert, you will use `messagebox`function `showinfo`. For creating a question, you will use the`askquestion` method, and based on the response to the question, you will output a `Label` on the GUI.

Rendering Images

If you have been able to follow along until here, then, adding images and icons to the GUI should be a piece of cake. All you need to do is use the `PhotoImage` method of



Tkinter and pass the `file_path` as the parameter to it.

Conclusion:-

It is my hope that this document will be of huge helpwith understanding of our little project as we have used a different approach which has proved beneficial for us and easy for us to understand the vast ocean that is Python.

References:-

To conduct this project the following tools have beenused :

- Pycharm and Github
- Tkinter (Library) : https://docs.python.org/3/library/tkinter.html
- Pillow (Library) : https://python-pillow.org/

1.1 Udemy:- I have used this site for our basis knowledge gain of the methods that will be used inthe project
https://www.udemy.com/course/100-days-of-code/

1.2 Stackoverflow:- I have used this site for solvingour different problems which have occurred during this project.

https://stackoverflow.com/questions/69467325/register-and-login-issue-using-tkinter