Python Functions

1. def function\_name(a,b) :- It creates a user defined function with or without positional arguments. E.g. otp\_generation()
2. string.digits :- This is a constant in the string module that provides a string of all the digits from 0 to 9.
3. ‘’”.join() :- This function is used in Python to concatenate or combine a sequence of strings into a single string, here string within (“ “) acts as separator to be added between two individual strings. E.g. “”.join(“a”, “b”, ”c”)
4. secrets.choice(sequence) :- This function selects one or more element from the sequence. Eg. secrets.choice(“a”,”b”,”c”,”d”)
5. range(n) :- This is a built-in function that generates a sequence of numbers, starting from 0 up to the number (n-1). It is commonly used in loops, especially in for loops, to iterate a specific number of times. Eg. range(6)
6. re.match(pattern, string) :- This regular expression function matches the pattern with the string at only start of the string. Eg. re.match(email\_pattern, receipient\_email)
7. Return :- This function returns the variable or function written after it. Eg. return recipient\_mail
8. smtplib.SMTP(host='', port=0, local\_hostname=None, [timeout, ]source\_address=None) This function is used to send a message from an email client to another email server. E.g. smtplib.SMTP("smtp.gmail.com", 587)
9. SMTP.starttls() :- The modern, flexible, and recommended standard for email transmission. It starts with a regular SMTP connection and upgrades to a secure, encrypted connection using STARTTLS. E.g. server.starttls()
10. SMTP.login(user, password, \*, initial\_response\_ok=True) :- The login() method is specifically used to authenticate to the email server using a username (usually the email address) and a password. E.g. server.login(sender\_email, sender\_password)
11. SMTP.sendmail(from\_addr, to\_addrs, msg, mail\_options=(), rcpt\_options=()) The sendmail method is specifically used to sent the “msg” from “from\_addr” to “to\_addr”. E.g. server.sendmail(sender\_email, receipient\_email, message)
12. server.quit() :- This method method is used to terminate the connection between the program and the SMTP server. After sending an email, it is a good practice to close the connection to free up resources and ensure the session is properly ended. E.g. server.quit()
13. global var\_name :- The global keyword is used to declare that a variable inside a function refers to a variable that is defined in the global scope, i.e., outside the function. By using global, you can modify a global variable inside a function, instead of just accessing its value.
14. time.time() :- It is a function from the built-in time module that returns the current time in seconds since the **epoch** (a fixed point in time, usually January 1, 1970 also known as Unix time)
15. windows.mainloop() :- This tkinter inbuild function starts the GUI event loop and the program enters an infinite loop that waits for events such as button clicks, key presses or other user interactions. E.g. Wind.mainloop()
16. tk.Tk() :- It is used to create the main window (also called the **root window**) for your graphical user interface (GUI) application. This is the first step when building a Tkinter application, as it provides the foundation upon which you add widgets like buttons, labels, text boxes, etc.
17. window.title("title\_name") :- This function is used to set the title of the window, which appears in the title bar at the top of the window. E.g. wind.title("OTP Verification System")
18. window.geometry("wxh") :- Creates GUI window with dimension in pixels with with w pixels and height h pixels. Eg. wind.geometry("300x210")
19. tk.Label(wind, text="label\_name") :- It is a widget used to display text or images on the window. It's one of the most basic and commonly used widgets in Tkinter for displaying information to the user, such as labels for input fields, static text, or even images. E.g. tk.Label(wind, text="Enter your Email ID :")
20. window.title("title\_name") :- This function is used to set the title of the window, which appears in the title bar at the top of the window. E.g. wind.title("OTP Verification System")
21. tk.Entry(parent\_window, option) :- It is a widget used to create a single-line text input field where users can type in information, such as text, numbers, or passwords. It’s one of the essential widgets in forms or any application where user input is required. E.g. tk.Entry(wind, width=30)
22. tk.Button(parent, text="Button Text", command=callback\_function) :- It is a widget used to create clickable buttons in a GUI application. Buttons are commonly used for user interaction, such as submitting data, triggering actions, or closing a window, here parameter command contains a function that is executed when the button is clicked E.g. tk.Button(wind,text="Send OTP",command=creating\_otp)
23. otp\_input.config(state='disabled’) :- this function is used to change the state of a widget, specifically an **Entry** widget (like a text input field), so that the user can no longer interact with it. This is useful when you want to prevent further input after the OTP has been entered or verified.
24. .get() :- This method is used with widgets like Entry, Text, and other similar input widgets to retrieve the current value or text that a user has entered. E.g. email\_input.get()
25. messagebox.showinfo() :- it is a function used to display an informational message to the user in a pop-up window. It is typically used to show some non-critical information, like a confirmation message, a success notification, or general information.
26. messagebox.showerror(title, message) :- This is used to display an error message to the user in a pop-up window. This function is useful when you want to alert the user to an error condition, such as invalid input or a failed operation. Eg. messagebox.showerror(“Error”, “Wrong OTP”)