# Code Explanation for Greeting App

## Overview

This document provides a detailed explanation of the Python code used in the Greeting App, which is built using Tkinter.

## Code Breakdown

### 1. Importing Required Modules

import tkinter as tk  
from tkinter import messagebox

• tkinter: Provides GUI functionalities.  
• messagebox: Used to display pop-up messages.

### 2. Creating the Main Window

root = tk.Tk()  
root.title("Greeting App")  
root.geometry("300x200")

• tk.Tk(): Initializes the application window.  
• title(): Sets the window title.  
• geometry(): Defines the window size.

### 3. Defining the Greeting Function

def greet():  
 name = name\_entry.get()  
 if name:  
 messagebox.showinfo("Greeting", f"Hello, {name}!")  
 else:  
 messagebox.showwarning("Input Error", "Please enter your name")

• name\_entry.get(): Retrieves user input from the text field.  
• messagebox.showinfo(): Displays a greeting message if a name is entered.  
• messagebox.showwarning(): Shows an error message if the input field is empty.

### 4. Adding UI Elements

#### Label for Name Input

name\_Label = tk.Label(root, text="Enter Your name:")  
name\_Label.pack(pady=10)

• Label(): Displays text prompting the user to enter their name.  
• pack(): Organizes the UI element with padding.

#### Entry Field for User Input

name\_entry = tk.Entry(root)  
name\_entry.pack(pady=10)

• Entry(): Creates an input field for the user.  
• pack(): Places the entry field with padding.

#### Button to Trigger Greeting

greet\_button = tk.Button(root, text="Greet", command=greet)  
greet\_button.pack(pady=10)

• Button(): Creates a button that calls the greet function.  
• pack(): Positions the button with padding.

### 5. Running the Application

root.mainloop()

• mainloop(): Starts the application's event loop, making the window interactive.

## Summary

The application collects user input, validates it, and displays a greeting message. Tkinter is used for building the graphical interface.