### Internship Project Report

#### Project Title: Development of To Do Task Application

#### Prepared By:

Ms.Aziza Azim Khalfe

#### Internship Position:

Web Development Intern

#### Company/Organization:

Unified Mentor

### Introduction

The purpose of this internship project was to design and develop a simple To-Do Task application using the Tkinter library in Python. This application allows users to manage their daily tasks efficiently by adding, removing, and marking tasks as completed.

### Objectives

* To understand the basics of GUI development using Tkinter.
* To implement core functionalities for a To-Do Task application.
* To provide a user-friendly interface for task management.

### Technologies Used

* **Programming Language**: Python
* **GUI Framework**: Tkinter

### Implementation

#### Code Overview

The following Python code snippet demonstrates the implementation of the To-Do Task application using Tkinter:

import tkinter as tk

from tkinter import messagebox

def add\_task():

task = entry.get()

if task:

listbox.insert(tk.END, task)

entry.delete(0, tk.END)

else:

messagebox.showwarning("Empty Bag")

def remove\_task():

selected\_task = listbox.curselection()

if selected\_task:

listbox.delete(selected\_task)

else:

messagebox.showwarning("No task selected", "Please select a task to remove.")

def mark\_completed():

selected\_task = listbox.curselection()

if selected\_task:

task = listbox.get(selected\_task)

listbox.itemconfig(selected\_task, {'bg': 'light green', 'selectbackground': 'light green'})

messagebox.showinfo("Task added", f"{task} marked as completed!")

else:

messagebox.showwarning("No Item Selected", "Please select an item to mark as completed.")

root = tk.Tk()

root.title("To Do Task App")

entry = tk.Entry(root, width=30)

entry.grid(row=0, column=0, padx=10, pady=10)

add\_button = tk.Button(root, text="Add Task", command=add\_task)

add\_button.grid(row=0, column=1, padx=5, pady=10)

listbox = tk.Listbox(root, selectmode=tk.SINGLE, height=10, width=30)

listbox.grid(row=1, column=0, columnspan=2, padx=10, pady=5)

remove\_button = tk.Button(root, text="Remove Task", command=remove\_task)

remove\_button.grid(row=2, column=0, pady=5)

complete\_button = tk.Button(root, text="Task Completed", command=mark\_completed)

complete\_button.grid(row=2, column=1, pady=5)

root.mainloop()

#### Detailed Explanation

**Importing Libraries:**

* 1. tkinter is imported for creating the GUI.
  2. messagebox is imported from tkinter for displaying message boxes.

**Function Definitions**:

* 1. add\_task(): Adds a task to the listbox. If the entry field is empty, a warning message is displayed.
  2. remove\_task(): Removes the selected task from the listbox. If no task is selected, a warning message is displayed.
  3. mark\_completed(): Marks the selected task as completed by changing its background color to light green. If no task is selected, a warning message is displayed.

**GUI Components**:

* 1. **Entry Widget**: Allows users to enter a new task.
  2. **Buttons**:
     1. add\_button to add a task.
     2. remove\_button to remove a task.
     3. complete\_button to mark a task as completed.
  3. **Listbox**: Displays the list of tasks.

**Event Loop**:

* 1. The root.mainloop() method starts the Tkinter event loop, which waits for user interaction.

### Conclusion

The To-Do Task application was successfully developed, providing a simple yet functional interface for managing daily tasks. The project helped in understanding the basics of GUI development with Tkinter and implementing core functionalities for a task management application.