Pomodoro Timer Project Report

Lavender-Themed Pomodoro Timer Using Python and Tkinter

Objective

To create a visually appealing desktop application based on the Pomodoro Technique using Python's Tkinter module.

Tools & Technologies

- Language: Python 3.x
- GUI Library: Tkinter
- Styling: Hex Colors (Lavender Theme)
- Notification: tkinter.messagebox
- Platform: Cross-platform (Windows/Linux/macOS)

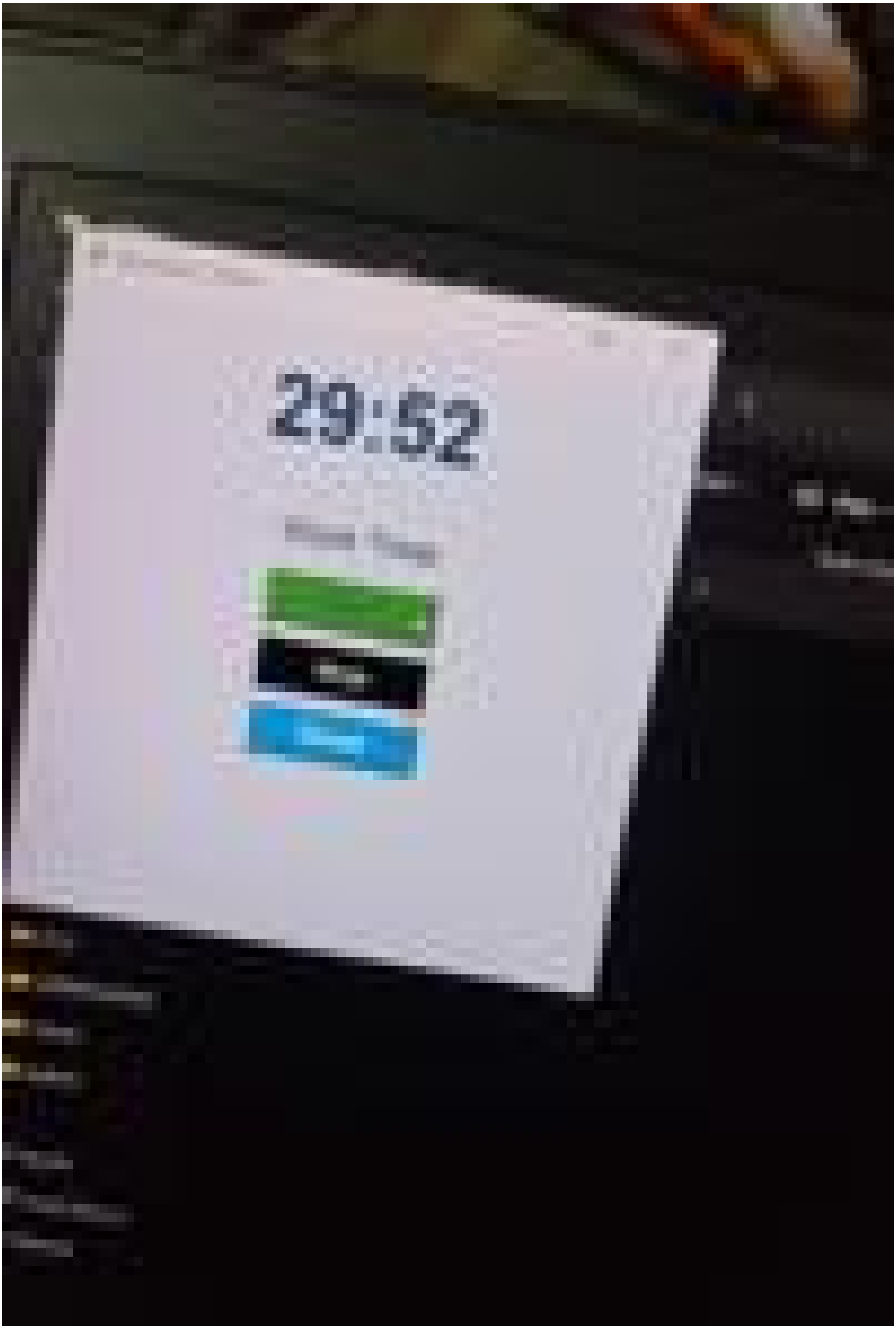
Pomodoro Technique

The Pomodoro Technique is a time management method where work is broken into 25-30 minute intervals separated by short breaks.

Features

- 30-minute work session
- 10-minute short break
- 20-minute long break after 4 Pomodoros
- Reset and pause functions
- Lavender-themed UI
- Pop-up alerts for transitions

Application Screenshot:



Source Code:

```
import tkinter as tk
from tkinter import messagebox

WORK_TIME = 30 * 60
SHORT_BREAK = 10 * 60
LONG_BREAK = 20 * 60

class PomodoroTimer:
    def __init__(self):
        self.root = tk.Tk()
        self.root.title("Pomodoro Timer")
        self.root.geometry("400x400")
        self.root.configure(bg="#E6E6FA")
        self.timer_label = tk.Label(self.root, text="30:00", font=("Arial", 40, "bold"), fg="#4B0082", bg="#F3E5F5")
        self.timer_label.pack(pady=30)
        self.mode_label = tk.Label(self.root, text="Work Time", font=("Arial", 16), fg="#2F004F", bg="#F3E5F5")
        self.mode_label.pack(pady=10)
        button_frame = tk.Frame(self.root, bg="#E6E6FA")
        button_frame.pack(pady=20)
        self.start_button = tk.Button(button_frame, text="Start", command=self.start_timer, width=10, bg="#9370DB", fg="white", font=("Arial", 12, "bold"))
        self.start_button.grid(row=0, column=0, padx=10)
        self.stop_button = tk.Button(button_frame, text="Stop", command=self.stop_timer, width=10, state=tk.DISABLED, bg="#6A5ACD", fg="white", font=("Arial", 12, "bold"))
        self.stop_button.grid(row=0, column=1, padx=10)
        self.reset_button = tk.Button(button_frame, text="Reset", command=self.reset_timer, width=10, bg="#8A2BE2", fg="white", font=("Arial", 12, "bold"))
        self.reset_button.grid(row=0, column=2, padx=10)
        self.work_time_left = WORK_TIME
        self.break_time_left = SHORT_BREAK
        self.is_work_time = True
        self.pomodoros_completed = 0
        self.is_running = False
        self.root.mainloop()

    def start_timer(self):
        if not self.is_running:
            self.is_running = True
            self.start_button.config(state=tk.DISABLED)
            self.stop_button.config(state=tk.NORMAL)
            self.update_timer()

    def stop_timer(self):
        self.is_running = False
        self.start_button.config(state=tk.NORMAL)
        self.stop_button.config(state=tk.DISABLED)

    def reset_timer(self):
        self.work_time_left = WORK_TIME
        self.break_time_left = SHORT_BREAK
        self.is_work_time = True
        self.is_running = False
        self.timer_label.config(text="30:00")
        self.mode_label.config(text="Work Time")
        self.start_button.config(state=tk.NORMAL)
        self.stop_button.config(state=tk.DISABLED)
    def update_timer(self):
        if not self.is_running:
            return
        if self.is_work_time:
            self.work_time_left -= 1
            if self.work_time_left <= 0:
                self.is_work_time = False
                self.pomodoros_completed += 1
                self.break_time_left = LONG_BREAK if self.pomodoros_completed % 4 == 0 else SHORT_BREAK
                messagebox.showinfo("Pomodoro Complete", "Time for a break!")
        else:
            self.break_time_left -= 1
            if self.break_time_left <= 0:
                self.is_work_time = True
                self.work_time_left = WORK_TIME
                messagebox.showinfo("Break Over", "Back to work!")
        current_time = self.work_time_left if self.is_work_time else self.break_time_left
        minutes, seconds = divmod(current_time, 60)
        self.timer_label.config(text=f"{minutes:02}:{seconds:02}")
        self.mode_label.config(text="Work Time" if self.is_work_time else "Break Time")
        self.root.after(1000, self.update_timer)

PomodoroTimer()
```