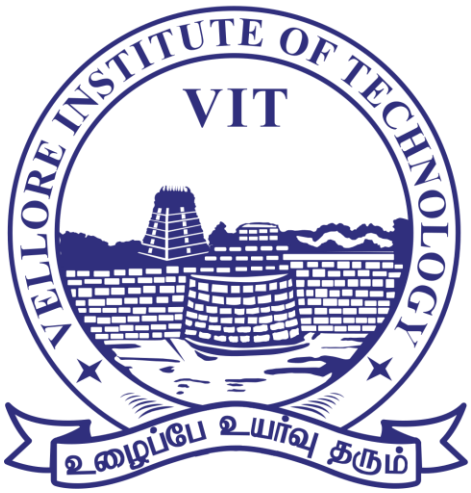


VIT®

BHOPAL

NAME: SHRIYA PARMAR

REG NO.: 25BAI10307

TOPIC: VITyarthi PROJECT

TITLE: MEDITATION APP(RESTAURA)

INTRODUCTION

Meditation is an effective practice for reducing stress, improving focus, and promoting mental well-being. With the increasing use of technology, meditation apps have become a convenient way to guide individuals through structured meditation sessions.

The **Meditation App** developed using **Python Tkinter** aims to provide users with a simple, interactive interface to play guided meditation sessions and soothing background hums, promoting relaxation and mindfulness.

PROBLEM STATEMENT

Many meditation apps are either complex or require subscription-based access. Users, especially beginners, often struggle to find lightweight tools that allow them to engage in meditation without distractions.

Problem Identified:

- Lack of offline, free meditation tools.
- Difficulty in integrating guided audio with relaxing background sounds.
- Absence of simple desktop-based solutions for mindfulness practice.

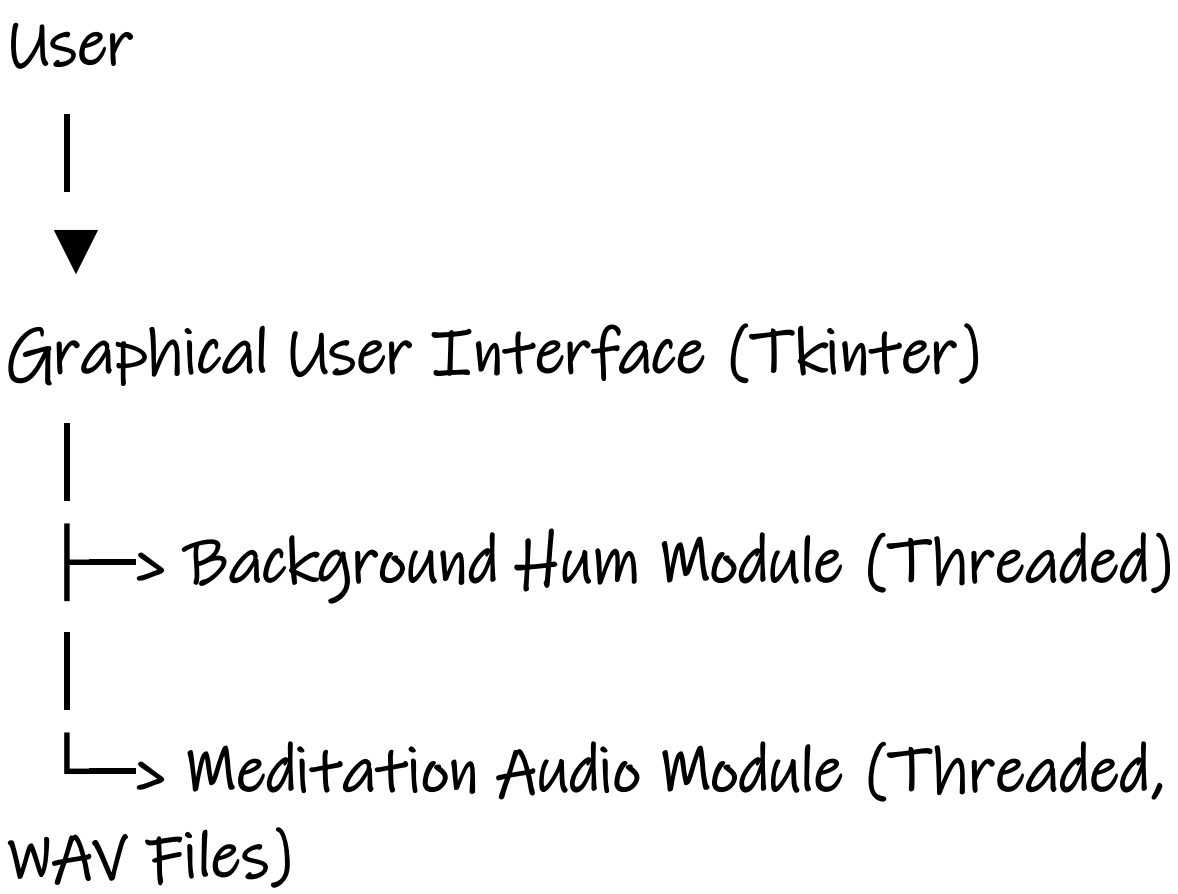
FUNCTIONAL REQUIREMENT

- User can **play guided meditation audio** files (e.g., breathing meditation, calming meditation).
- User can **start and stop a background hum** to enhance relaxation.
- User can **stop all audio** at once.
- Simple **Graphical User Interface (GUI)** for easy navigation.
- **Multi-threading** to ensure background hum and meditation audio play simultaneously without freezing the GUI.

NON-FUNCTIONAL REQUIREMENTS

- **Performance:** Audio playback should be smooth without delays.
- **Usability:** GUI should be intuitive and simple.
- **Portability:** Works on **Windows OS** using Python and Tkinter.
- **Reliability:** Handles stop/start commands gracefully without crashing.
- **Maintainability:** Modular design for adding new meditations easily.

ARCHITECTURE OVERVIEW



 The system uses **threading** to allow simultaneous audio playback and UI responsiveness.

 Tkinter handles all GUI components.

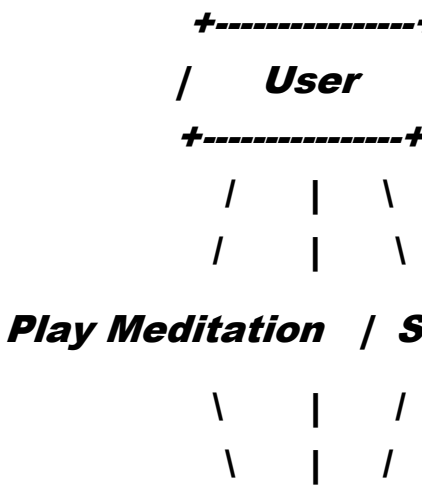
 Winsound library is used for audio playback

DESIGN OVERVIEW

- ## USE CASE DIAGRAM

Actors: User

Use Cases: Play Meditation, Start/Stop Background Hum, Stop All Audio



- ## WORK FLOW DIAGRAM

Start App

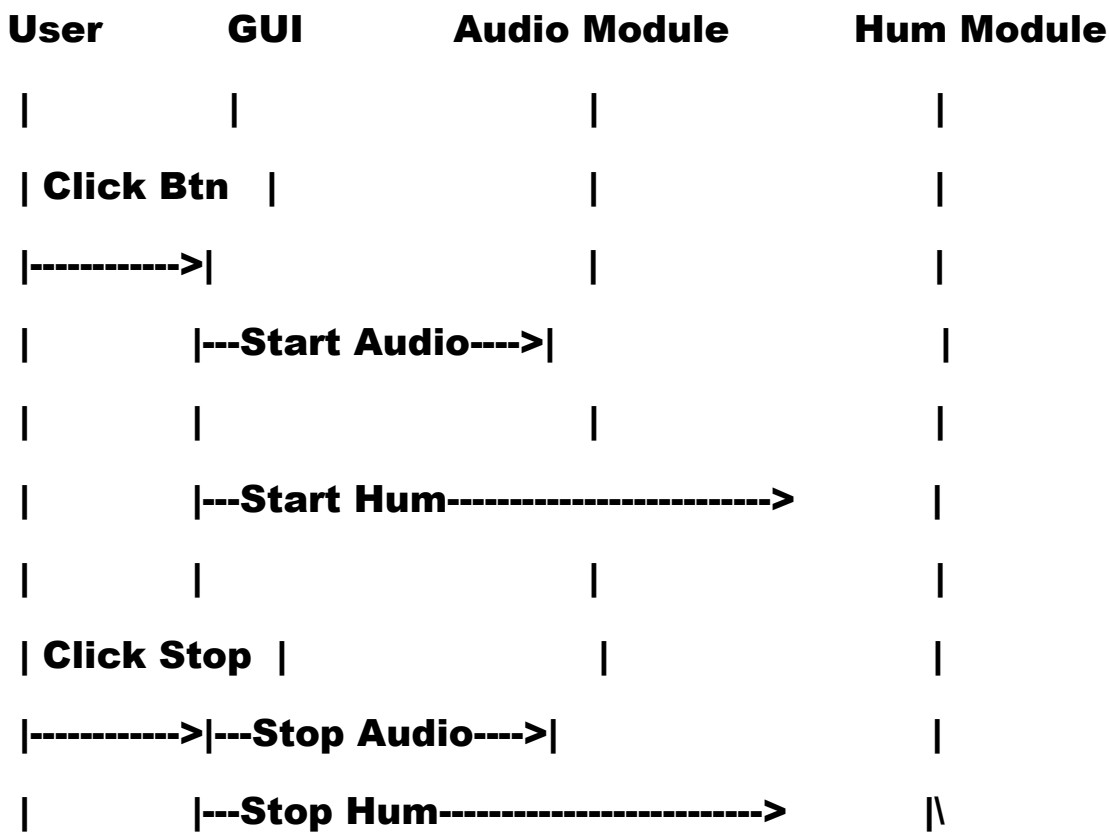
Display GUI with buttons

├—> User selects meditation → Play selected audio

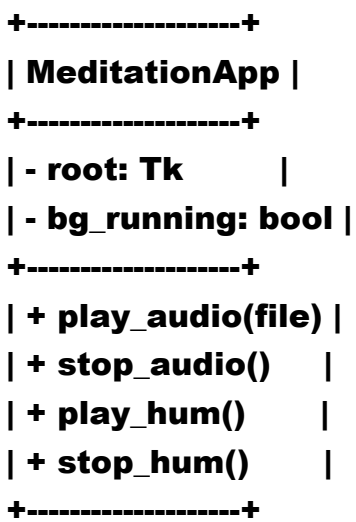
├—> User starts background hum → Hum plays in loop

└—> User stops all audio → Stop meditation + stop hum

- SEQUENCE DIAGRAM



- CLASS DIAGRAM



.DESIGN DECISIONS & RATIONALE

- **Tkinter GUI**: Lightweight, native Python library for desktop apps.
- **Winsound Module**: Simplifies audio playback without external dependencies.
- **Threading**: Ensures the UI remains responsive while audio plays in the background.
- **Simple Buttons Layout**: Allows users to start/stop sessions easily.

IMPLEMENTATION DETAILS

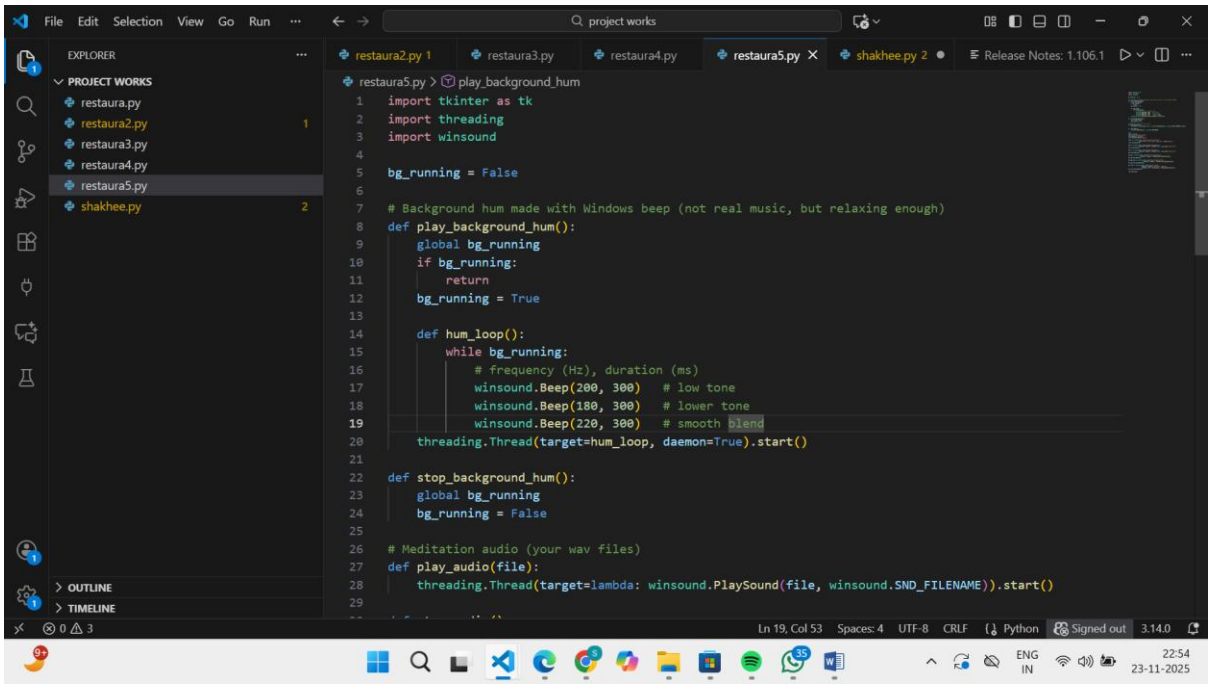
- **Language & Libraries:** Python 3.x, Tkinter, Winsound, Threading.
- **Audio Files:** WAV format for compatibility with Winsound.
- **Background Hum Logic:** Looping low-frequency beeps to simulate a calming hum.
- **Threading:** Each audio component runs in a separate thread for smooth operation.

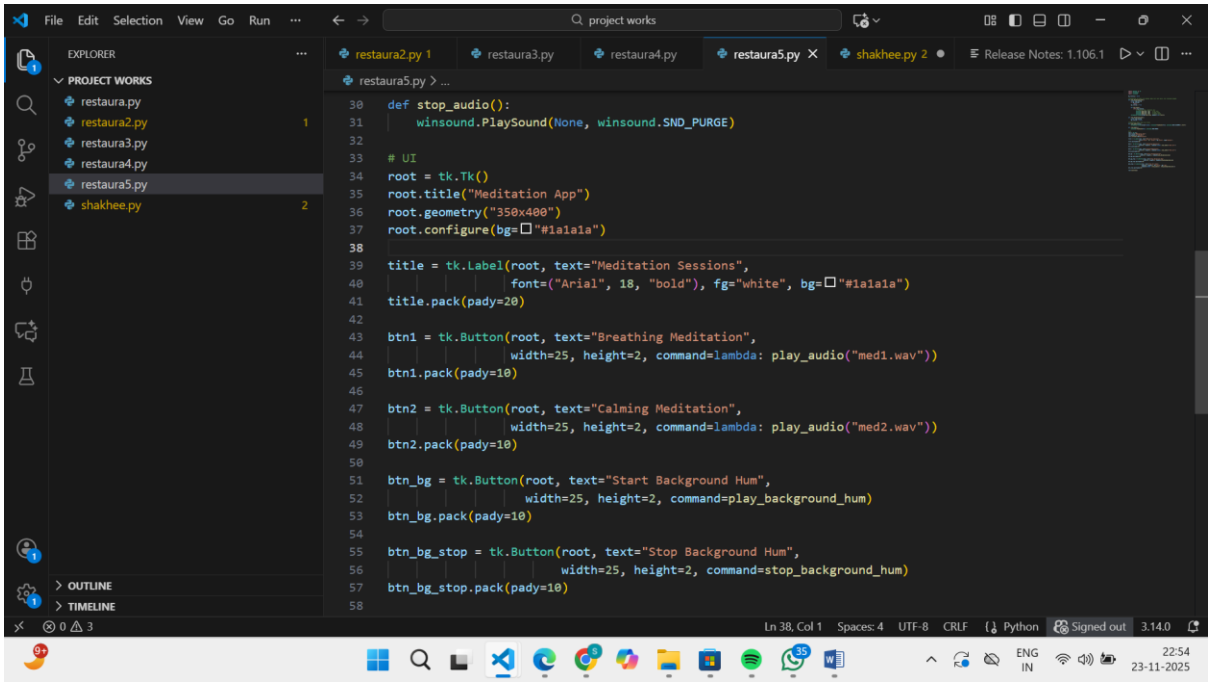
Sample code snippet

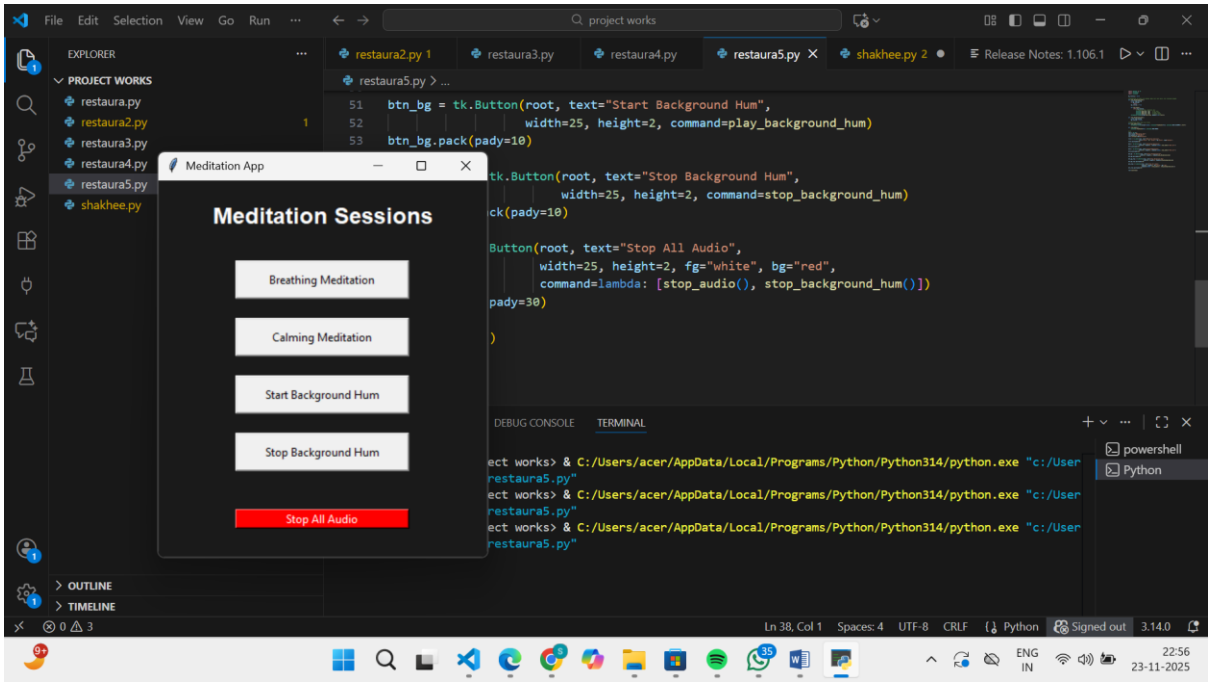
```
import tkinter as tk

import threading

import winsound


bg_running = False


def play_background_hum():
```

```
global bg_running
if bg_running:
    return
bg_running = True
def hum_loop():
    while bg_running:
        winsound.Beep(200, 300)
        winsound.Beep(180, 300)
        winsound.Beep(220, 300)
threading.Thread(target=hum_loop, daemon=True).start()
```

SCREENSHOTS OF CODE




File Edit Selection View Go Run
project works
EXPLORER
PROJECT WORKS
restaura.py
restaura2.py 1
restaura3.py
restaura4.py
restaura5.py
shakhee.py 2
restaura2.py 1
restaura3.py
restaura4.py
restaura5.py
shakhee.py 2
Release Notes: 1.106.1
restaura5.py > ...
30 def stop_audio():
31 winsound.PlaySound(None, winsound.SND_PURGE)
32
33 # UI
34 root = tk.Tk()
35 root.title("Meditation App")
36 root.geometry("350x400")
37 root.configure(bg="#1a1a1a")
38
39 title = tk.Label(root, text="Meditation Sessions",
40 font=("Arial", 18, "bold"), fg="white", bg="#1a1a1a")
41 title.pack(pady=20)
42
43 btn1 = tk.Button(root, text="Breathing Meditation",
44 width=25, height=2, command=lambda: play_audio("med1.wav"))
45 btn1.pack(pady=10)
46
47 btn2 = tk.Button(root, text="Calming Meditation",
48 width=25, height=2, command=lambda: play_audio("med2.wav"))
49 btn2.pack(pady=10)
50
51 btn_bg = tk.Button(root, text="Start Background Hum",
52 width=25, height=2, command=play_background_hum)
53 btn_bg.pack(pady=10)
54
55 btn_bg_stop = tk.Button(root, text="Stop Background Hum",
56 width=25, height=2, command=stop_background_hum)
57 btn_bg_stop.pack(pady=10)
58
OUTLINE
TIMELINE
Ln 38, Col 1 Spaces: 4 UTF-8 CRLF Python Signed out 3.14.0
ENG IN
22:54
23-11-2025


File Edit Selection View Go Run
project works
EXPLORER
PROJECT WORKS
restaura.py
restaura2.py 1
restaura3.py
restaura4.py
restaura5.py
shakhee.py
restaura2.py 1
restaura3.py
restaura4.py
restaura5.py
shakhee.py 2
Release Notes: 1.106.1
restaura5.py > ...
51 btn_bg = tk.Button(root, text="Start Background Hum",
52 width=25, height=2, command=play_background_hum)
53 btn_bg.pack(pady=10)
tk.Button(root, text="Stop Background Hum",
width=25, height=2, command=stop_background_hum)
ck(pady=10)
Button(root, text="Stop All Audio",
width=25, height=2, fg="white", bg="red",
command=lambda: [stop_audio(), stop_background_hum()])
pady=30)
)
Meditation App
Meditation Sessions
Breathing Meditation
Calming Meditation
Start Background Hum
Stop Background Hum
Stop All Audio
DEBUG CONSOLE TERMINAL
ect works> & C:/Users/acer/AppData/Local/Programs/Python/Python314/python.exe "c:/User
restaura5.py"
ect works> & C:/Users/acer/AppData/Local/Programs/Python/Python314/python.exe "c:/User
restaura5.py"
ect works> & C:/Users/acer/AppData/Local/Programs/Python/Python314/python.exe "c:/User
restaura5.py"
powershell
Python
OUTLINE
TIMELINE
Ln 38, Col 1 Spaces: 4 UTF-8 CRLF Python Signed out 3.14.0
ENG IN
22:56
23-11-2025

TESTING APPROACH

- **Functional Testing:** Tested all buttons to ensure audio plays/stops correctly.
- **Threading Testing:** Verified that starting hum and meditation audio simultaneously does not freeze GUI.
- **Usability Testing:** Checked if buttons are intuitive for new users.

CHALLENGES FACED

- Handling **thread safety** while starting and stopping audio threads.
- Limitation of **Winsound library** (supports only WAV files).
- Ensuring GUI **remains responsive** during audio playback.

LEARNINGS & KEY TAKEAWAYS

- Learned **multi-threading in Python** to handle concurrent tasks.
- Gained experience in **GUI design** using Tkinter.
- Understood **audio handling in Python** and its limitations.
- Learned to plan a project report including **UML diagrams** and system documentation.

FUTURE ENHANCEMENTS

- Integrate **MP3 support** using pygame or pydub.
- Add a **timer and reminders** for meditation sessions.
- Implement **progress tracking** and **statistics** for regular users

- Include **customizable background sounds** (nature, rain, wind, etc.).
- Expand to **cross-platform support** beyond Windows.

REFERENCES