Python Alarm Clock (GUI) – Project Report

1. Introduction

This project is a graphical Alarm Clock application built using Python's built-in **Tkinter** library. It displays a real-time digital clock and allows the user to set an alarm using a simple input field. The application plays a sound and shows a popup notification when the alarm time is reached. It is designed as an easy and practical beginner Python GUI project.

---

2. Objective

The main objectives of this project are: - To understand GUI development using Tkinter - To work with time and date using Python - To learn how to run background tasks without freezing the UI - To build a functional alarm with sound and notifications

---

3. Tools & Technologies Used

- **Python 3.x**
- **Tkinter** (GUI framework)
- **datetime** (to fetch current time)
- **time** (for delays)
- **winsound** (for alarm sound on Windows)
- **threading** (for non-blocking alarm playback)

---

4. System Requirements

- Python 3 installed
- Windows/Linux/macOS
- No external modules required

---

5. Working Principle

The GUI alarm clock works through the following steps: 1. The clock displays current system time, updated every second. 2. User enters alarm time in **HH:MM** or **HH:MM:SS** format. 3. Application stores the alarm time and checks every second if it matches the current time. 4. When the time matches: - A popup message appears - Alarm sound plays using a background thread 5. User may stop or cancel the alarm anytime.

---

6. User Interface Description

The UI contains: - **Digital Clock Display** (HH:MM:SS) - **Input field** for alarm time - **Set Alarm** button - **Stop Alarm** button - **Status label** showing current alarm status

All elements are created using Tkinter widgets such as `Label`, `Entry`, `Button`, and `Frame`.

---

7. Source Code

The complete Python GUI code is saved in the file `alarm_gui.py` which includes: - Tkinter window setup - Time updater function - Alarm setter and stopper - Sound playback - Background threading to avoid freezing the GUI

(Refer to the code file in this project.)

---

8. Output (Expected Behavior)

When the application runs: 1. A window opens showing the clock and alarm input box. 2. After entering a valid alarm time and clicking **Set Alarm**, a confirmation message appears. 3. When the system time matches the alarm time: - A popup window appears saying *"Time's up!"* - Alarm sound plays repeatedly 4. Alarm status resets automatically after ringing.

---

9. Applications of This Project

- Personal reminder system
- Daily alarm utility
- Useful for learning GUI design
- Demonstrates real-time data handling
- Great as a Python mini-project for assignments

---

10. Limitations

- MP3 sounds are not supported by default
- Lacks snooze/multiple alarms
- Basic UI without themes
- Minimal error checking for invalid formats

---

11. Future Enhancements

- Add MP3 music playback
- Add snooze button
- Add multiple alarm scheduling
- Add dark/light themes
- Add animated UI
- Save alarms permanently using a database

---

12. Conclusion

This project demonstrates how to build a practical and fully functional alarm clock using Python’s Tkinter GUI framework. It helps beginners learn about interface design, time handling, and multi-threaded execution in a simple and effective way.

Prepared By: Amit Kumar Pandey**