

# **Project Report: Medication Reminder & Pill Identification System**

## **1. Introduction**

Managing medication schedules can be difficult, especially for individuals who are taking multiple medicines or following strict timing requirements. Missing a dose or taking medication inconsistently can affect both treatment outcomes and overall health.

This project was created to offer a simple, reliable, and offline solution for users who want a tool that does not depend on external libraries, accounts, or internet connectivity. The goal was to build a small but functional command-line application that helps users stay organized with their daily medications while also offering a basic pill identification feature.

---

## **2. Project Overview**

The Medication Reminder & Pill Identification System is a Python-based CLI application designed to support daily medication management. It provides three core functionalities:

### **1. Medication Tracking:**

Users can add medications, specify dosage, frequency, and timing, and store this data for long-term use.

### **2. Reminder Service:**

The program continuously monitors the system time and notifies the user when it is time to take a medication.

### **3. Pill Identification:**

A small built-in database allows users to identify pills based on color, shape, and imprint code.

The system is meant to be simple, practical, and easy to run on any machine with Python installed.

---

### **3. Objectives**

The primary objectives of the project were:

- To create a lightweight medication reminder tool using only standard Python libraries.
  - To ensure medication data is persistently stored between sessions.
  - To implement a background reminder service that requires minimal user intervention.
  - To incorporate a small-scale pill identification module that can assist with basic pill matching.
  - To build the system in a way that is easy to understand, modify, and extend.
- 

## **4. System Design**

### **4.1 Architecture**

The application revolves around a single class, **MedicationReminder**, which handles all major functionalities:

- Managing medication data (adding, viewing, marking as taken)
- Interacting with the medications.json file
- Running the reminder loop
- Identifying pills from the internal database

The structure is intentionally simple so that beginners can understand the flow of data and the logic behind each function.

## 4.2 Storage

A JSON file (medications.json) is used to store all medication entries. This avoids the need for external databases while still keeping data persistent and easy to read or edit.

## 4.3 Reminder Mechanism

The reminder system runs an infinite loop which checks the current time every 60 seconds. If a medication's scheduled time matches the system time and it hasn't been taken for the day, the program prompts the user.

## 4.4 Pill Identification Module

A predefined dictionary acts as the pill database. Each pill entry includes basic attributes such as color, shape, and imprint. The system attempts to match user input with the stored attributes and returns possible matches.

---

## 5. Features

### 5.1 Add Medication

Users can enter key information about a medication—name, dosage, frequency, and schedule. The program validates time input and stores all data automatically.

## **5.2 View Medications**

Outputs a clean summary of all stored medications, including their last taken date and active status.

## **5.3 Mark Medication as Taken**

Allows users to manually confirm they have taken their medication for the day. This prevents repeated reminders.

## **5.4 Pill Identification**

Provides possible pill matches based on user-entered characteristics.

## **5.5 Reminder Service**

Continuously runs in the background and notifies the user when scheduled medication times are reached.

---

## **6. Limitations**

While the system is functional, it has a few limitations:

- The reminder only works while the script is running.
- The pill database is intentionally small and not intended for medical use.
- There are no push notifications or sound alerts.
- Time matching is exact (HH:MM), without tolerance handling.

These limitations can be addressed in future versions.

---

## 7. Future Improvements

Potential enhancements include:

- Adding sound or desktop notifications.
  - Expanding the pill database or integrating external APIs (e.g., FDA NDC).
  - Introducing a GUI for more user-friendly interactions.
  - Adding medication history tracking and analytics.
  - Allowing flexible scheduling (e.g., every 8 hours).
  - Adding profile support for multiple users.
- 

## 8. Conclusion

The Medication Reminder & Pill Identification System is a practical, easy-to-use tool developed using basic Python components. It demonstrates how small-scale applications can still be meaningful and helpful in everyday life. By combining medication management, basic pill identification, and persistent data storage, the project offers a reliable utility while remaining simple enough for beginners to understand, modify, or expand.

This project also provides a foundation for future development and can evolve into a robust personal healthcare assistant with additional features and refinements.

