Smart Alarm Clock Project – Rapberry pi 5 version 0.0

Project Overview:

The Alarm Clock project is a Python-based application designed to allow users to set and manage alarms using a web interface. The application utilizes Flask, a micro web framework, for handling HTTP requests and providing a user interface. Users can set alarms by specifying the date, hour, minute, and optionally upload a custom ringtone. Alarms can be toggled on or off, and the system will trigger the alarms at the specified times.

Features:

Web Interface: Users can interact with the alarm clock application through a web interface. They can set alarms, toggle alarm status, and view upcoming alarms.

Alarm Setting: Users can set alarms by specifying the date, hour, and minute for the alarm to trigger. Additionally, they can upload custom ringtone files to be played when the alarm triggers.

Alarm Toggling: Users can toggle the status of alarms on or off. This allows them to enable or disable specific alarms without deleting them.

Bluetooth Integration: The application integrates with Bluetooth-enabled devices to stream audio from mobile devices. It can connect to a specified Bluetooth device and stream custom ringtone audio when alarms trigger.

Multi-threaded Operation: The application utilizes threading to handle concurrent operations. One thread manages the Flask web server, while another thread handles alarm logic, allowing for seamless interaction and alarm triggering.

Implementation Details:

The application is implemented in Python, utilizing libraries such as Flask for web server functionality and subprocess for executing system commands.

Alarms are stored in a dictionary, allowing for easy retrieval and modification of alarm details.

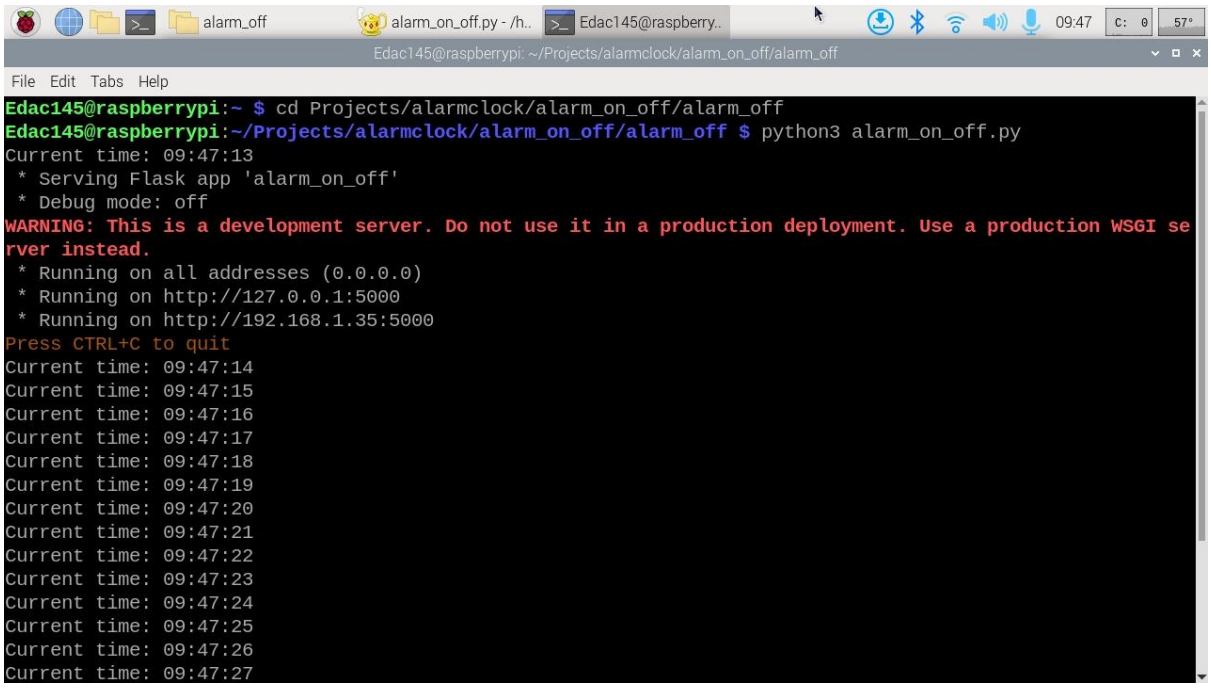
The system checks the current time continuously, comparing it with the set alarm times to trigger alarms when conditions are met.

Custom ringtone files uploaded by users are saved to a specified directory for later use when alarms trigger.

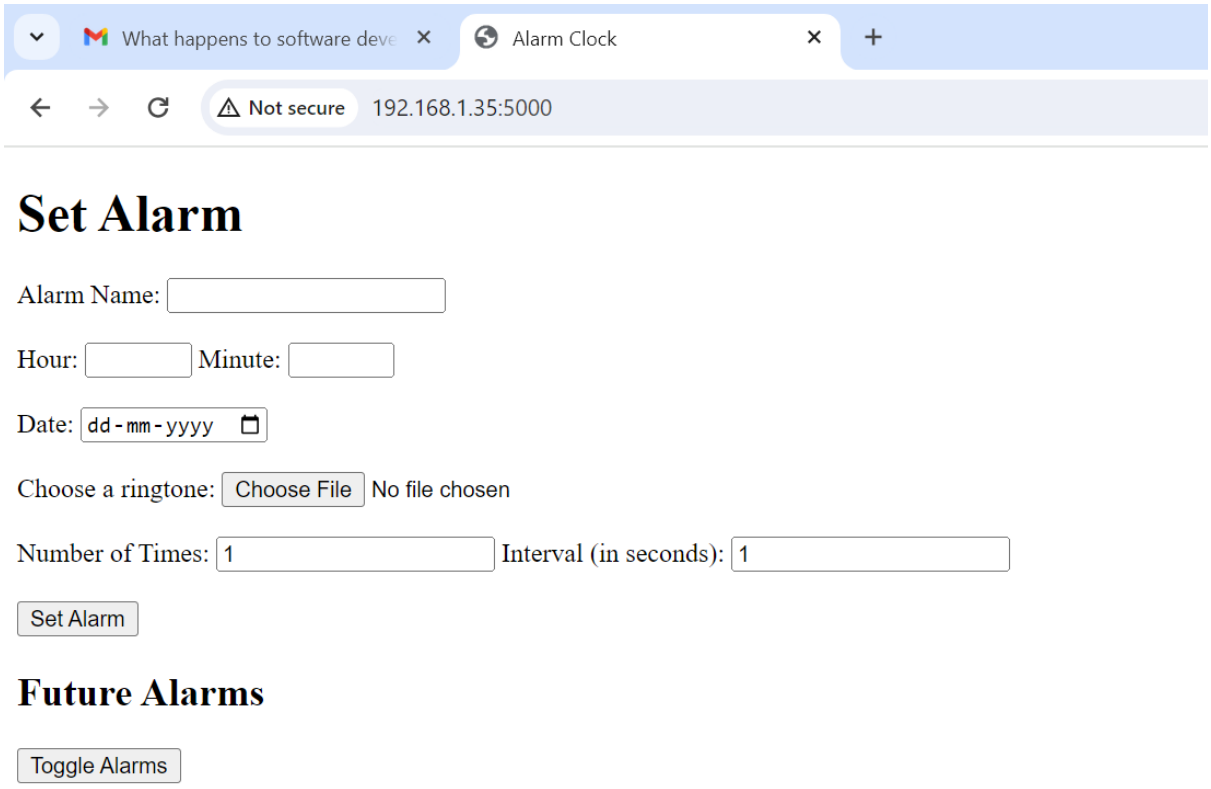
Bluetooth functionality is used to connect to a Bluetooth-enabled device and stream audio for alarm notifications and entertainment.

Usage:

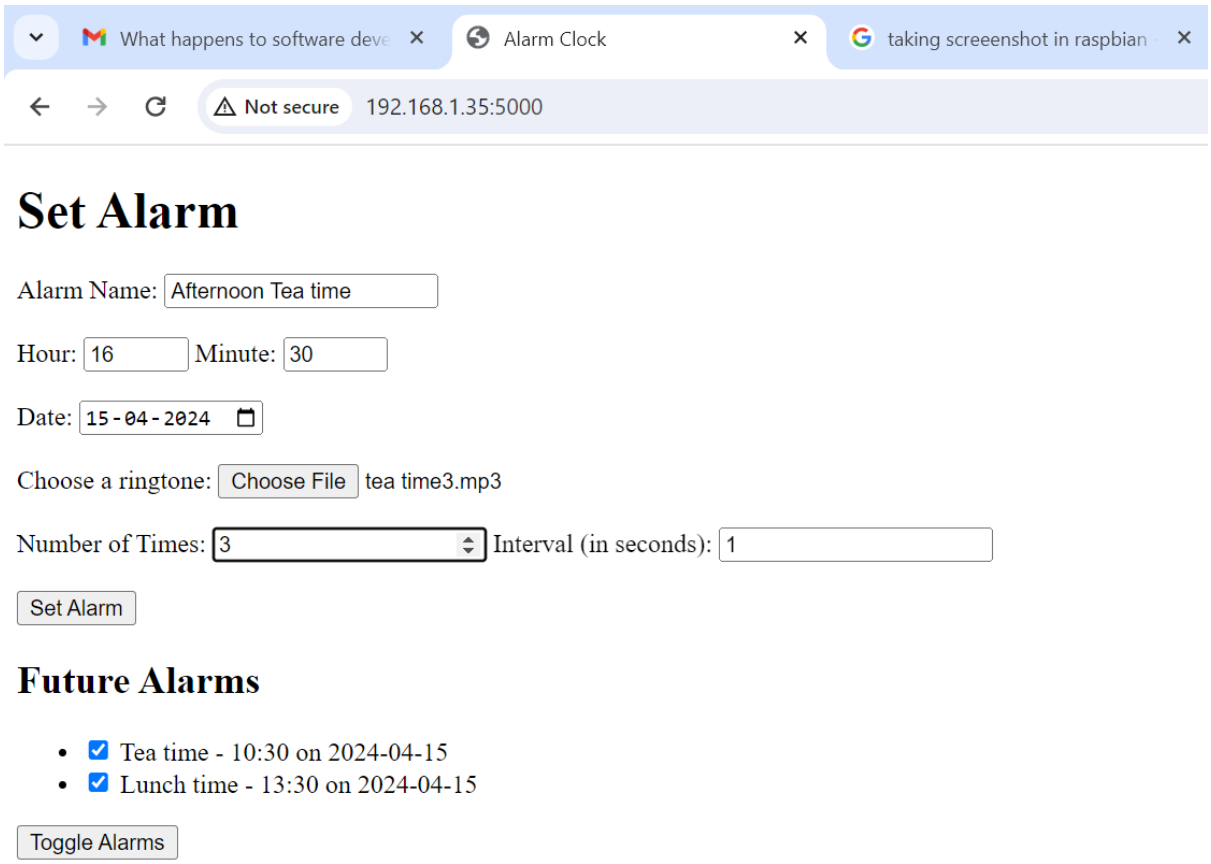
1. Run the Python script to start the alarm clock application.



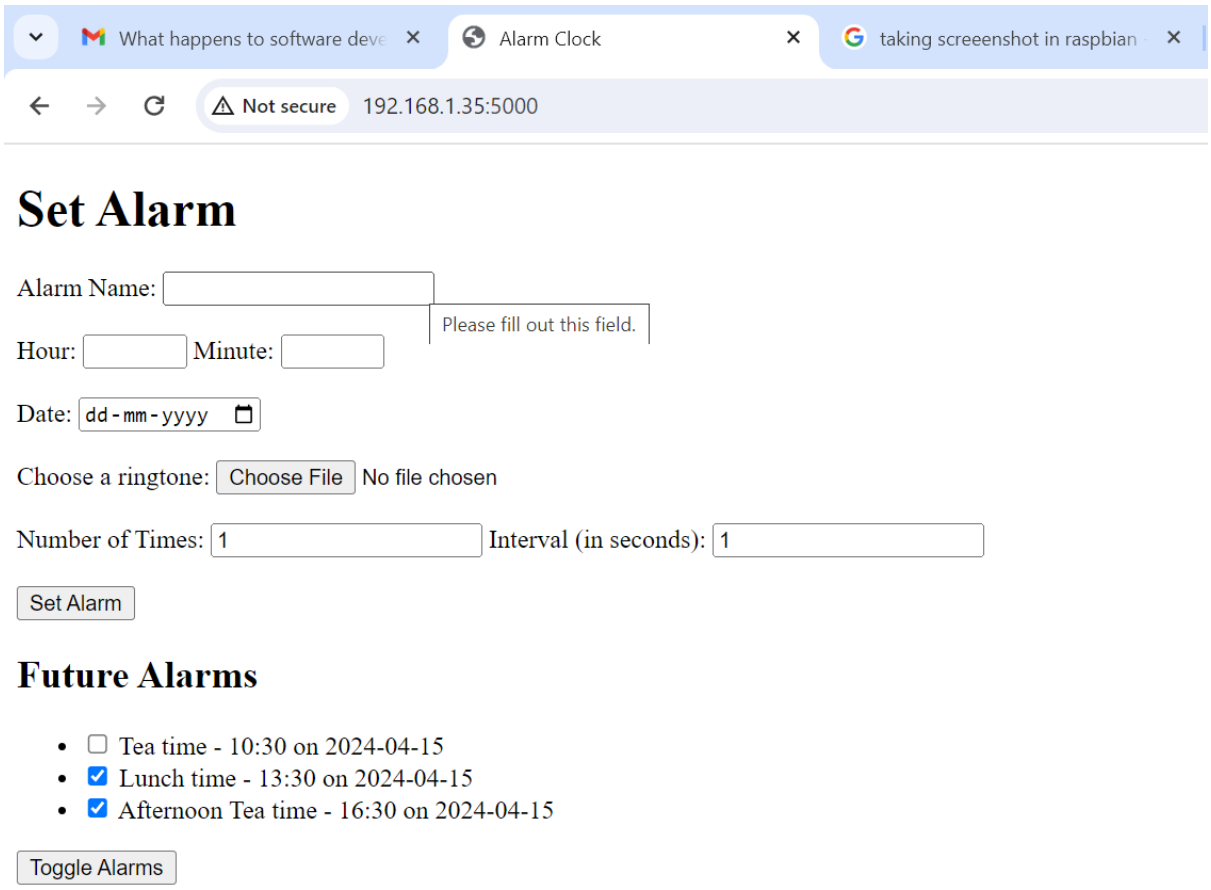
2. Access the web interface through a web browser.



3. Set alarms by specifying the date, hour, minute, and optional ringtone file upload.



4. Toggle alarm status as needed.



The system will trigger alarms at the specified times, playing the custom ringtone audio through the connected Bluetooth device.

Future Improvements:

Enhance user interface design for improved user experience.

Implement user authentication and session management for secure access.

Add support for recurring alarms (e.g., daily, weekly).

Integrate with additional smart devices for enhanced functionality

Note: This document provides an overview of the Alarm Clock project, including its features, implementation details, and usage instructions. Further enhancements and modifications can be made based on specific requirements and feedback from users.