# **Smart House Automation System**

**Objective:**

This project aims to develop a SMART HOME AUTOMATION system using Python. It enables users to remotely control and automate various home devices, as well as log device data for analysis and visualization.

**Table of content**

* Initialization
* Project Description
* Features
* Insights
* Dependencies
* Tools Used
* Usage
* Links
* Author Name
* **Initialization :**

1. **Install python**: Ensure the python 3x is installed to your computer you can download it from [Official Python Website](https://www.python.org/downloads/)
2. **Install libraries:**

* Numpy
* Pandas
* Matplotlib
* Seaborn
* Datetime
* **Project Description:**

This project involves creating a smart home management system using Python. The system is designed using OOP principles, where a base class Device is created with common attributes and methods. Subclasses for specific devices like Thermostat, Light, and Security Camera are derived from the base class, each with additional specific attributes.

File handling functionality is implemented to save and load device data. Conditional logic controls device operations based on specific conditions, such as time-based lighting control and temperature-based thermostat adjustments.

Device data is logged periodically and stored in a Pandas Data Frame for analysis. Data visualizations are created using Matplotlib to provide insights into device usage and performance.

* **Features**
  + Object-Oriented Design:
    - Base Class (Device): The foundational class includes common attributes such as device id, device type, and status, along with methods for retrieving and setting the device status.
    - Subclasses:
    - Thermostat: Inherits from Device, adds attributes for temperature settings.
    - Light: Inherits from Device, adds attributes for brightness levels.
    - Security Camera: Inherits from Device, adds attributes for recording status.
  + File Handling:
    - Saving Data: Device data is saved to files in JSON or CSV format, ensuring data persistence.
    - Loading Data: Functionality to load device data from files allows for easy restoration and continuity of device states.
  + Conditional Logic:
    - Time-Based Control: Implementing logic to turn on lights based on the time of day, such as turning lights on between sunset and sunrise.
    - Temperature-Based Control: Adjusting thermostat settings based on real-time temperature readings.
  + Data Logging and Analysis:
    - Periodic Logging: Device data, such as temperature readings, is logged periodically and stored in a Pandas Data Frame.
    - Data Analysis: Using Pandas to analyse logged data, providing insights like average temperature over a period and identifying peak energy usage times.
  + Data Visualization:
    - Line Charts: Display temperature changes over time.
    - Bar Charts: Show energy usage by device.
    - Pie Charts: Illustrate the proportion of time each device is active.
* **Dependencies:**

To run this project, the following libraries are required:

- Python 3.x: The core programming language.

- Pandas: For data storage, manipulation, and analysis.

- Matplotlib: For creating visualizations of device data.

- JSON: For saving and loading device data.

- CSV: For handling CSV files for device data.

Install the dependencies using:

bash

pip install numpy pandas matplotlib seaborn

* **Insights:**
  + Device Usage Patterns: Analysis of logged data can reveal detailed usage patterns, such as peak energy usage times, helping optimize device settings and reduce energy consumption.
  + Performance Monitoring: Continuous monitoring and analysis of device data enable the identification of performance issues or unusual activity, ensuring timely maintenance and security.
  + User Behaviour Analysis: Insights into user interactions with devices can inform system improvements, enhancing user experience and system efficiency.
* **Tools Used:**
* Python: Used as the core programming language for implementing the system due to its versatility and rich ecosystem.
* Pandas: Utilized for data storage, manipulation, and analysis.
* Matplotlib: Used for creating visualizations of the logged data.
* Seaborn: Used for creating advanced and attractive statistical visualizations.
* NumPy: Used for generating random values.
* datetime: Used for handling date and time operations.
* JSON/CSV: Formats for saving and loading device data, ensuring ease of data handling and compatibility.
* **Usage:**
* Run the Script: Ensure all dependencies are installed. Then, run your Python script (e.g., smart\_home.py) using the terminal.
* Data Input: Load data from JSON or CSV files as needed using Pandas.
* Data Manipulation: Use Pandas to manipulate and analyse the data.
* Generating Random Values: Use NumPy to generate random values for simulations or testing.
* Date and Time Operations: Use the datetime module to handle date and time operations.
* Data Visualization: Use Matplotlib and Seaborn to create visualizations.
* Saving Data: Save the manipulated data back to a CSV or JSON file.
* **Exploratory Data Analysis (EDA)**
* **Summary:**
* Temperature Trends: Observed temperature variations throughout the day and shift, with average, maximum, and minimum values recorded.
* Brightness Patterns: Analysed brightness levels over time, indicating the intensity of illumination.
* Light Usage: Identified the frequency of light usage, distinguishing between on and off states.
* **Insights:**
* Environmental Conditions: Temperature and brightness data provide valuable insights into environmental conditions, aiding in understanding daily and shift-wise variations.
* Operational Efficiency: Analysis of light usage patterns helps assess operational efficiency and energy consumption.
* Monitoring Security: Correlation between temperature and security camera activity can offer insights into security monitoring effectiveness.
* **Visualizations:**
* Temperature over Time
* Brightness Distribution
* Security Camera Over Time
* Temperature Distribution
* Energy Usage by Device
* Brightness Over Time
* Frequency of Each Thermostat Action
* Proportion of Time Each Device is On
* Temperature by Security Camera
* **Links:**
* [**Pandas Documentation**](https://pandas.pydata.org/pandas-docs/stable/)
* [**Numpy Documentation**](https://numpy.org/devdocs/user/)
* [**Matplotlib Documentation**](https://matplotlib.org/stable/contents.html)
* [**Seaborn Documentation**](https://seaborn.pydata.org/)
* [**Datetime Documentation**](https://docs.python.org/3/library/datetime.html)
* [**CSV Documentation**](https://docs.python.org/3/library/csv.html)
* **Conclusion**

In conclusion, the analysis of temperature, brightness, and light usage data provides valuable insights into environmental conditions, operational efficiency, and security monitoring. By examining temperature trends and brightness patterns, we gain a deeper understanding of daily and shift-wise variations, aiding in informed decision-making.

The analysis also sheds light on the frequency of light usage, helping organizations assess their energy consumption and operational efficiency. Furthermore, the correlation between temperature and security camera activity underscores the potential for leveraging environmental data for enhanced security monitoring.

* **Future Works**

Moving forward, several avenues for future exploration and improvement present themselves:

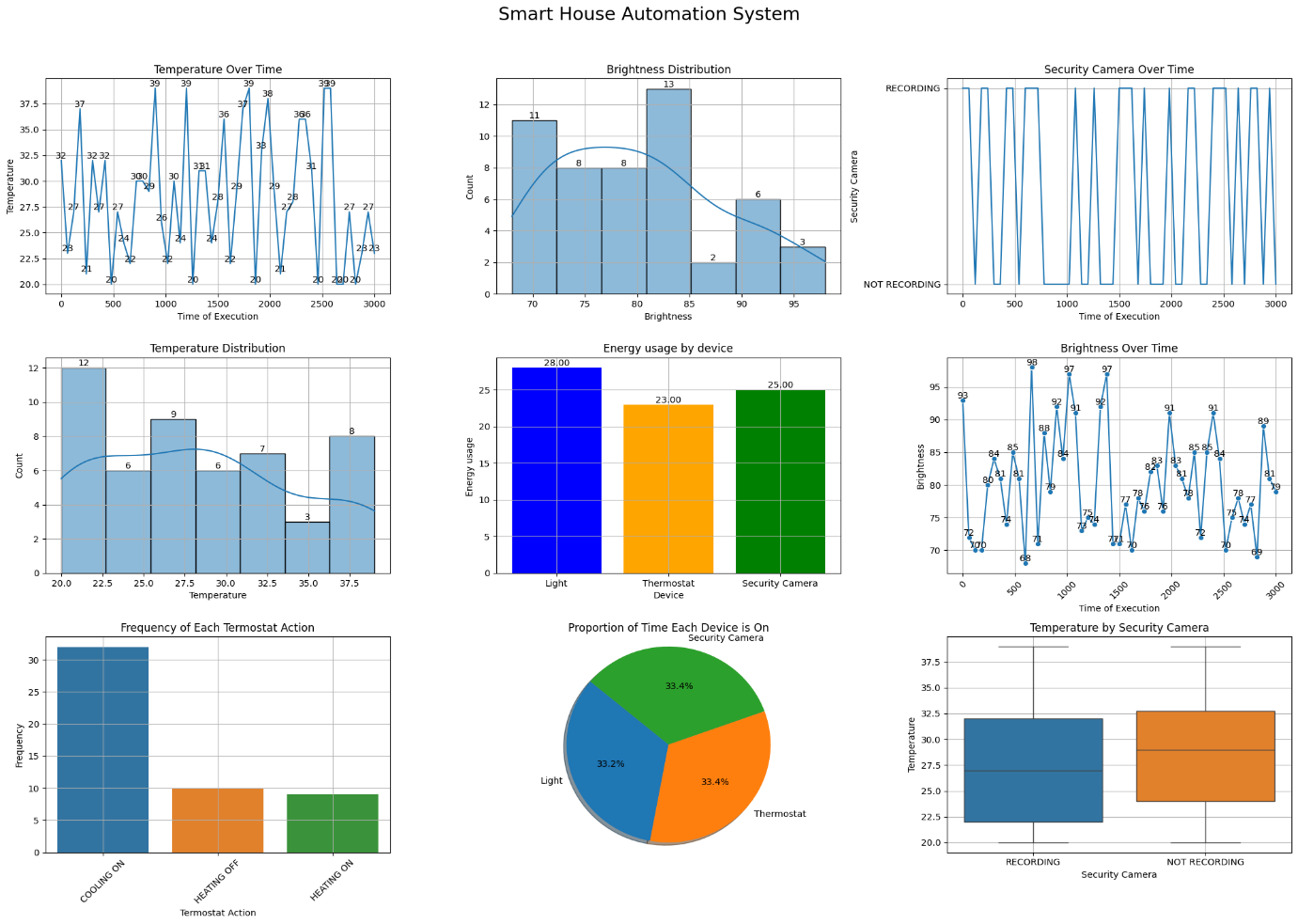
Predictive Modeling: Develop predictive models to forecast temperature trends and brightness patterns, enabling proactive decision-making.

Optimization Strategies: Implement optimization strategies based on light usage data to enhance energy efficiency and reduce operational costs.

Integration with IoT Devices: Integrate environmental data with IoT devices for real-time monitoring and automated control systems.

Enhanced Security Monitoring: Explore advanced analytics techniques to further enhance security monitoring capabilities, such as anomaly detection and event prediction.

* **Output:**

****

* **Author Name:** Praneeth M
* **Mobile number:** 938143119
* **Email:** [mpraneeth.vertocity@gmail.com](mailto:mpraneeth.vertocity@gmail.com)