REPORT - INT 301

Open Source Technologies

B.Tech. CSE

Submitted to - Rajeshwar Sharma Sir

PROJECT- **Using Open Source Software to *System information utility , *Display the Mac product key and ID, *A list of installed softwares, *and all the currently running processes , *GITHUB-Repository , *And save the Report in a simple text file.**

Submitted by-

1. **Nishant kumar**

Department of Computer science and engineering LPU

1.Introduction -

1.1_Objective of the project

- **To develop a system information utility using open source software.**
- **To display the Windows product key and ID.**
- **To list installed softwares .**
- **To list all the currently running processes.**
- **To save the report in a simple text file.**
- **To upload the report to the GITHUB-Repository.**

1.2_Description of the project

The aim of this project is to develop a system information utility that can be used to display various information about a Windows computer. The utility will be developed using open source software to ensure that it is easily accessible and can be used by anyone.

The utility will be designed to display the Windows product key and ID, a list of all installed software on the computer, and a list of all the currently running processes. The information will be displayed in a user-friendly format that is easy to read and understand.

The utility will also include a feature to save the report in a simple text file. This will allow users to easily access the report later, without having to run the utility again.

1.2_Scope of the project

The scope of this project is limited to developing a system information utility that can display the Windows product key and ID, a list of installed software, and all the currently running processes. The utility will be developed using open source software and will be designed to save the report in a simple text file.

2.SYSTEM DESCRIPTION -

2.1_Target system description

The scope of this project is limited to developing a system information utility that can display the Windows product key and ID, a list of installed software, and all the currently running processes. The utility will be developed using open source software and will be designed to save the report in a simple text file. The project does not aim to provide any additional functionality beyond the scope of the defined objectives.

2.2_Assumptions and dependencies

- **The project assumes that the user has basic knowledge of using command-line interfaces.**
- **The project depends on the Homebrew package manager for macOS to install and manage command-line software.**
- **The project depends on the "mas" command-line tool, which is a package provided by Homebrew, to access the Mac App Store and retrieve information about installed applications.**
- **The project assumes that the user has a stable internet connection to install and update the Homebrew package manager and the “mas” command-line tool.**

2.3_Functional dependencies

- **The project relies on the "system_profiler" command-line tool to gather system information utility, such as the Mac product key and ID and a list of installed software.**
- **The project relies on the "ps" command-line tool to retrieve information about all currently running processes on the Mac computer.**

2.4 Non-Functional dependencies

- **The project should be efficient and responsive, providing system information utility and process information in a timely manner.**
- **The project should be user-friendly, providing clear and concise instructions for the user to access system information utility and process information.**
- **The project should be reliable and accurate, providing correct and up-to-date information about the system.**

2.5 Data-set used in support of your project

The project does not rely on any external datasets. However, it uses the information provided by the "system_profiler" and "ps" command-line tools to gather and display system information utility and process information.

3.Analysis Report-

STEP 1: Install Homebrew:

Homebrew is an open source software. It is a popular package manager for macOS that allows users to easily install and manage open source software packages and libraries from the command line. The Homebrew codebase is available on GitHub and is distributed under the MIT license, which means that it is free to use, modify, and distribute.
Homebrew is a package manager for macOS that provides a simple and efficient way to install, manage, and update open source software packages and libraries from the command line. With Homebrew, users can easily install and manage a wide range of software packages and libraries, including programming languages, web servers, databases, command-line tools, and more.

Some of the benefits of using Homebrew include:

- **Easy installation and management of open source software packages and libraries**
- **Automatic dependency resolution and installation**
- **Ability to update packages with a single command**
- **Support for multiple versions of the same package**
- **Simple uninstallation of packages**
- **Integration with macOS and the command line interface**

Overall, Homebrew makes it easy for users to set up and maintain a custom software development environment on macOS, with access to a wide range of open source software packages and libraries.

Open Terminal and run the following command to install Homebrew:

/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"

STEP 2: Install the “mas” command-line tool using Homebrew:

‘brew install mas’

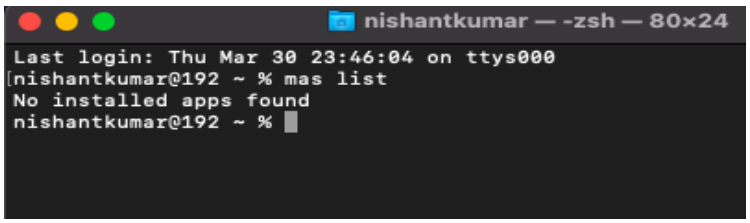
This command will install the "mas" tool, which is a command-line interface for the Mac App Store.

STEP 3: To list all the installed applications from the App Store, run the following command in Terminal:(*Apps installed from Apple store only)

'mas list'

This will display a list of all the apps installed on your Mac that were downloaded from the App Store.

SNAPSHOTS

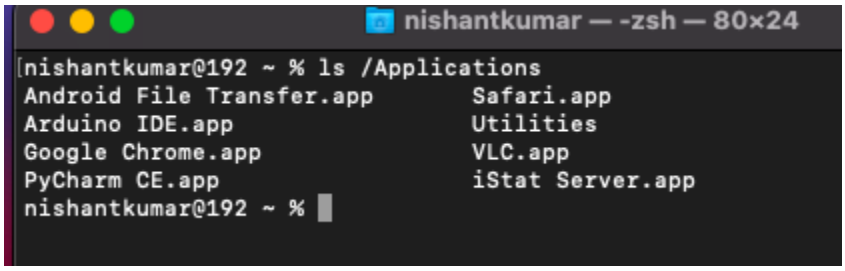


Since I have not installed any apps from the apple store it's showing no installed apps found.

STEP 4: To list all the installed applications installed on mac including both third-party apps and system apps run the following command in the terminal:

'ls /Applications'

SNAPSHOTS

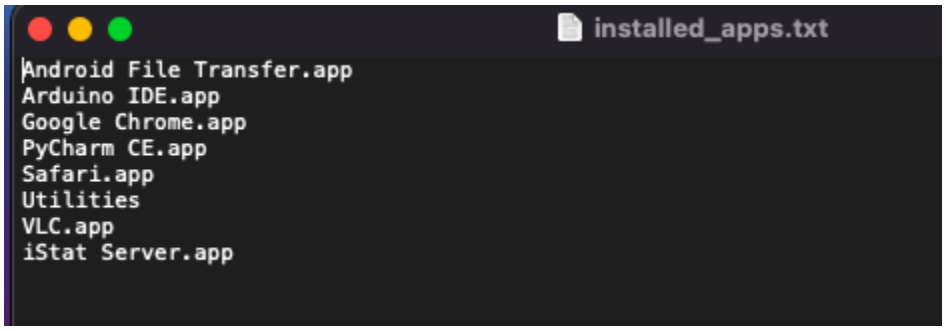


STEP 5:To save the list of installed applications to a text file run the following command in the terminal:

'ls /Applications > installed_apps.txt'

This will save the output of the “ls /Applications” command to a file named “installed_apps.txt.”

SNAPSHOTS

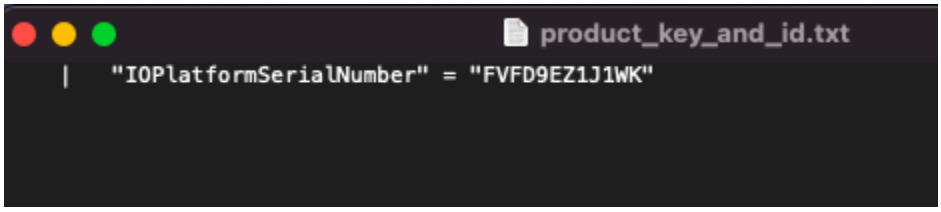


STEP 6: To get the product key and ID for your Mac and saving it in a file , run the following command in Terminal:

‘ioreg -l | grep IOPlatformSerialNumber > product_key_and_id.txt’

This will save the product key and ID information to a file named "product_key.txt" in the current directory.

SNAPSHOTS

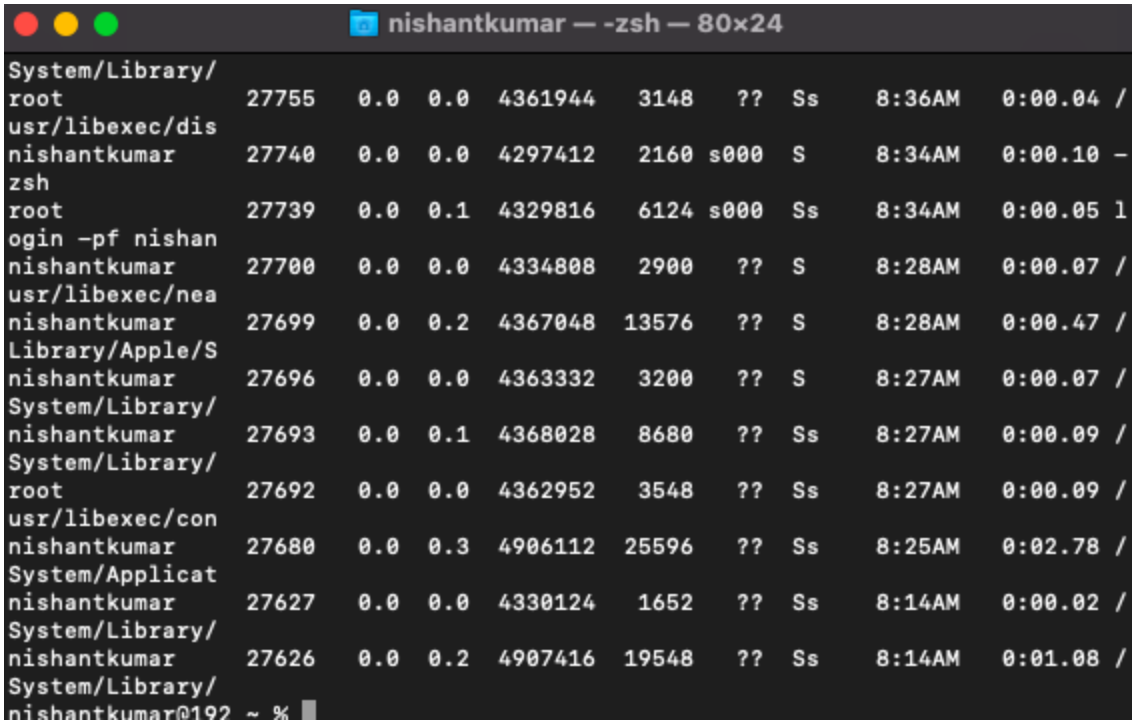


STEP 7: To list all the currently running processes, run the following command in Terminal:

‘ps aux’

This will list all the running processes and display them on the terminal.

SNAPSHOTS



STEP 8:To save all the running processes to a txt file run the following command in the terminal:

‘ps aux > running_processes.txt’

This will save the list of running processes to a file named "running_processes.txt" in the current directory.

SNAPSHOTS

```
running_processes.txt
USER               PID  %CPU %MEM      VSZ    RSS   TT  STAT STARTED      TIME COMMAND
nishantkumar     27075   2.7  0.1  4338140  11000   ??  S    11:55PM   0:00.10 /System/Library/
Frameworks/CoreServices.framework/Frameworks/Metadata.framework/Versions/A/Support/mdworker_shared -s
mdworker -c MDSImporterWorker -m com.apple.mdworker.shared
nishantkumar     35944   1.9  3.1 39056428 262056   ??  S    10Mar23 360:48.10 /Applications/Google
Chrome.app/Contents/MacOS/Google Chrome
nishantkumar     16308   0.7  3.7 1190578372 310880   ??  S     6:40AM  10:32.06 /Applications/Google
Chrome.app/Contents/Frameworks/Google Chrome Framework.framework/Versions/110.0.5481.177/Helpers/
Google Chrome Helper (Renderer).app/Contents/MacOS/Google Chrome Helper (Renderer) --type=renderer --
lang=en-GB --num-raster-threads=2 --enable-zero-copy --enable-gpu-memory-buffer-compositor-resources
--enable-main-frame-before-activation --renderer-client-id=6397 --time-ticks-at-unix-
epoch=-1675283000259111 --launch-time-ticks=3773292744628 --shared-files --field-trial-
handle=1718379636,r,5317984445530243184,5913093049791061096,131072 --seatbelt-client=375
root               262   0.5  0.4  6080024  31104   ??  Ss   23Aug22  58:28.05 /System/Library/
Frameworks/CoreServices.framework/Frameworks/Metadata.framework/Versions/A/Support/mds_stores
root                96   0.4  0.2  4482628  13496   ??  Ss   23Aug22  80:35.11 /System/Library/
Frameworks/CoreServices.framework/Frameworks/Metadata.framework/Support/mds
nishantkumar     27054   0.1  0.1  4338140  11180   ??  S    11:53PM   0:00.12 /System/Library/
Frameworks/CoreServices.framework/Frameworks/Metadata.framework/Versions/A/Support/mdworker_shared -s
mdworker -c MDSImporterWorker -m com.apple.mdworker.shared
nishantkumar       313   0.1  0.0  4331228   2684   ??  S    23Aug22  12:39.74 /usr/sbin/cfprefsd
agent
nishantkumar     35956   0.1  1.0 38641220  86736   ??  S    10Mar23 789:52.20 /Applications/Google
Chrome.app/Contents/Frameworks/Google Chrome Framework.framework/Versions/110.0.5481.177/Helpers/
Google Chrome Helper (GPU).app/Contents/MacOS/Google Chrome Helper (GPU) --type=gpu-process --gpu-
preferences=UAAAAAAAAAAAgAAAIAAAAAAAAAAAAAAAAAABgAAAAAAAwAAAAAAAAAAAAAAAQAAAAAAAAAAAAAAAAAAAAAAAAALgFA
AAAAAAAuAUAAAAAAADoAQAAPAAAAOABAAAAAAAA6AEAAAAAAADwAQAAAAAAAPgBAAAAAAAAAAIAAAAAAAAIAgAAAAAAABACAAAAAA
AAGAIAAAAAAAAgAgAAAAAAACgCAAAAAAAAMAIAAAAAAAA4AgAAAAAAAEACAAAAAAAASAIAAAAAAABQAgAAAAAAAFgCAAAAAAAAYAI
AAAAAAABoAgAAAAAAAHACAAAAAAAAeAIAAAAAAACAAgAAAAAAAIgCAAAAAAAAkAIAAAAAAACYAgAAAAAAAKACAAAAAAAAqAIAAAAA
AACwAgAAAAAAALgCAAAAAAAAwAIAAAAAAADIAgAAAAAAANACAAAAAAAA2AIAAAAAAADgAgAAAAAAAOgCAAAAAAA8AIAAAAAAAD4A
```

STEP 9:To display system information utility (HARDWARE) run the following command in the terminal:

'system_profiler SPHardwareDataType'

This will display information about your system hardware, such as the processor, memory, and storage.

SNAPSHOTS

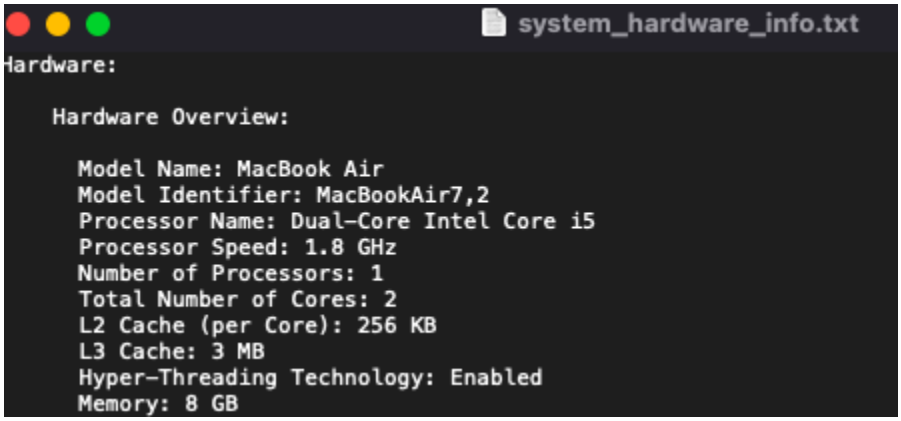
```
Documents — -zsh — 80×24
Overview:
macOS
Protection:
[nishantkumar@192 Documents % system_profiler SPHardwareDataType
Hardware:

    Hardware Overview:

      Model Name: MacBook Air
      Model Identifier: MacBookAir7,2
      Processor Name: Dual-Core Intel Core i5
      Processor Speed: 1.8 GHz
      Number of Processors: 1
      Total Number of Cores: 2
      L2 Cache (per Core): 256 KB
      L3 Cache: 3 MB
      Hyper-Threading Technology: Enabled
```

STEP 10:To save system information utility (HARDWARE) in a text file run the following command in the terminal:

‘system_profiler SPHardwareDataType | tee system_hardware_info.txt’

This will save the output of the “system_profiler” command to a file named “system_hardware_info.txt”

SNAPSHOTS

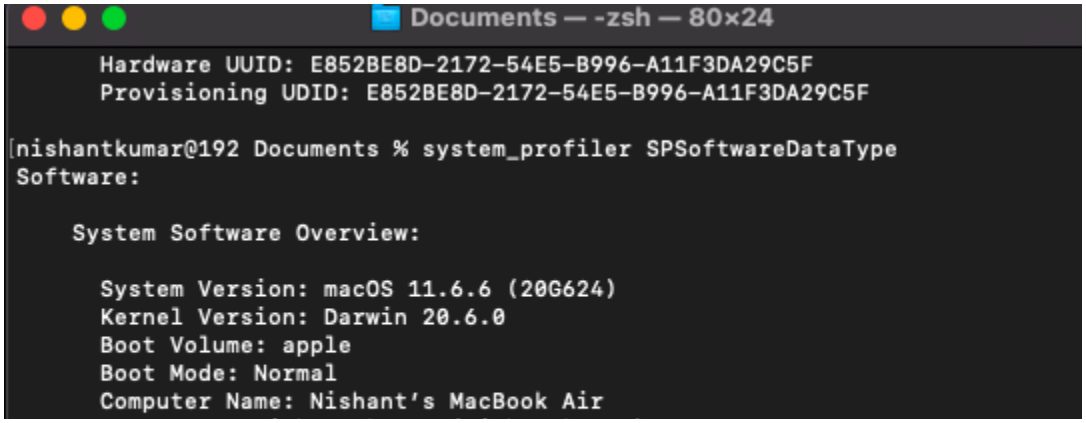


STEP 11:To display system information utility (SOFTWARE) run the following command in the terminal:

‘system_profiler SPSoftwareDataType’

This will display information about your system software, such as the operating system version and installed applications.

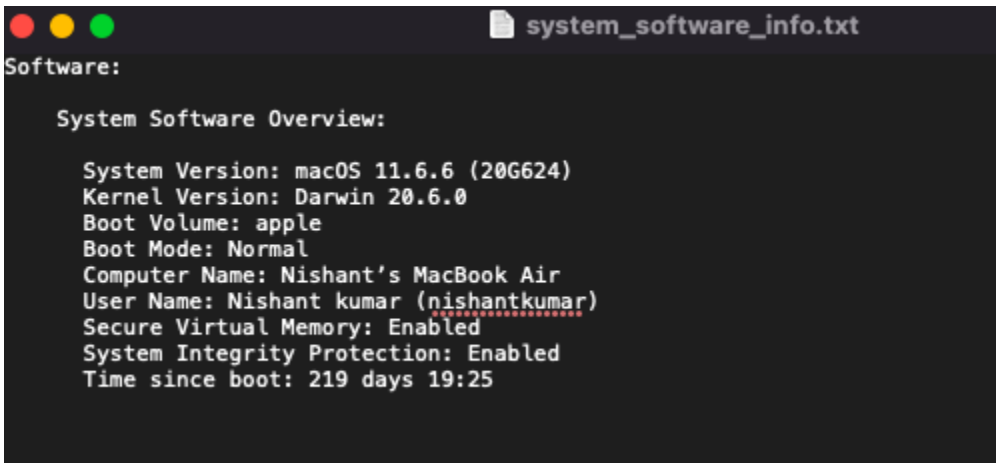
SNAPSHOTS



STEP 12:To save system information utility (SOFTWARE) in a text file run the following command in the terminal:

‘system_profiler SPSoftwareDataType | tee system_software_info.txt’

SNAPSHOTS



STEP 13: To save all these processes in the final report txt file . Displaying system information utility , Displaying the mac product key and id , Displaying the list of installed from 3rd party aswell and DisplayING all the running processes.

‘cat system_info.txt product_key.txt installed_apps.txt running_processes.txt > final_report.txt’

This command will concatenate the contents of the four text files I have created earlier into a single file named "final_report.txt". The ">" symbol is used to redirect the output of the "cat" command to the file named "final_report.txt".

SNAPSHOTS

final_report.txt

```
Accessibility:

    Accessibility Information:

      Cursor Magnification: Off
      Display: Black on White
      Flash Screen: Off
      Mouse Keys: Off
      Slow Keys: Off
      Sticky Keys: Off
      VoiceOver: Off
      Zoom Mode: Full Screen
      Contrast: 0
      Keyboard Zoom: Off
      Scroll Zoom: Off

Applications:

    iStat Server:

      Version: 3.03
      Obtained from: Identified Developer
      Last Modified: 03/08/17, 3:02 AM
      Kind: Intel
      Signed by: Developer ID Application: Bjango Pty Ltd (Y93TK974AT), Developer ID Certification
Authority, Apple Root CA
      Location: /Applications/iStat Server.app

    Python Launcher 3:
```

final_report.txt

```
      Signed by: Developer ID Application: JetBrains s.r.o. (2ZEFAR8TH3), Developer ID Certification
Authority, Apple Root CA
      Location: /Applications/PyCharm CE.app
      Get Info String: PyCharm 2022.2.1, build PC-222.3739.56. Copyright JetBrains s.r.o., (c)
2000-2022

    XProtect:

      Version: 93
      Obtained from: Apple
      Last Modified: 17/03/23, 6:25 PM
      Kind: Universal
      Signed by: Software Signing, Apple Code Signing Certification Authority, Apple Root CA
      Location: /Library/Apple/System/Library/CoreServices/XProtect.app

    VLC:

      Version: 3.0.17.3
      Obtained from: Identified Developer
      Last Modified: 12/06/22, 4:39 PM
      Kind: Intel
      Signed by: Developer ID Application: VideoLAN (75GAHG3SZQ), Developer ID Certification
Authority, Apple Root CA
      Location: /Applications/VLC.app

    Android File Transfer:

      Version: 1.0.12
      Obtained from: Identified Developer
      Last Modified: 16/10/18, 11:38 AM
```

PROJECT OUTCOMES- **Tasks that can be performed by a home automation system using Raspberry Pi include:**

1. Controlling lights: The system can be used to turn lights on and off, or to dim them to a desired level.

2. Controlling fans: The system can be used to turn fans on and off, or to adjust their speed.

3. Controlling air conditioners: The system can be used to turn air conditioners on and off, or to adjust their temperature.

4. Monitoring temperature and humidity: The system can be used to monitor the temperature and humidity in various rooms and adjust them as needed.

5. Detecting gas leaks: The system can be used to detect gas leaks and alert the occupants of the house.

6. Detecting intruders: The system can be used to detect intruders and alert the occupants of the house.

7. **Controlling home appliances**: The system can be used to control various home appliances, such as ovens, refrigerators, and washing machines.

Overall, home automation using Raspberry Pi is a great way to make your home smarter and more efficient, while also

HOME AUTOMATION -

Home automation using Raspberry Pi is a popular project that involves controlling various appliances and devices in a house or building through a central system. This system can be controlled remotely using a smartphone, tablet or computer. Here are some of the components and tasks that can be included in a home automation project using Raspberry Pi:

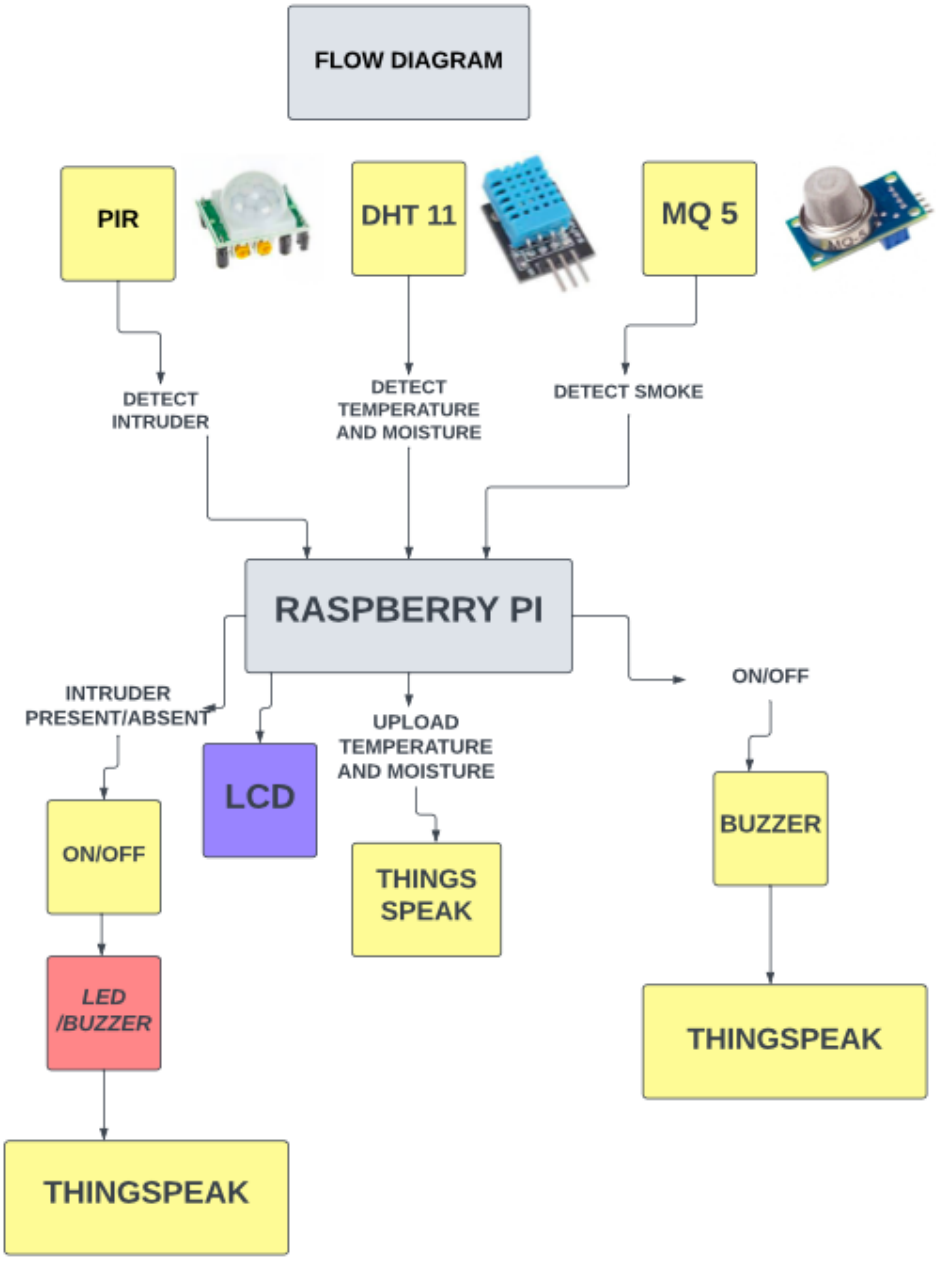
1. **Hardware Requirements**: To start with, we will need a Raspberry Pi board, a power supply, an SD card, sensors and actuators, and relays. we will also need jumper wires and a breadboard to connect the sensors and actuators.
2. **Connecting the sensors and actuators**: we can connect the sensors and actuators to the Raspberry Pi using jumper wires and a breadboard. For example, we can connect a motion sensor to a GPIO pin on the Pi, and connect an LED or a relay to another GPIO pin.
3. **Writing the code**: we will need to write code in a programming language Python to control the sensors and actuators. we will use libraries like RPi.GPIO to interface with the GPIO pins on the Pi.
4. **Testing the system**: Once we have set up the circuit and written the code, we will test the system by accessing the web interface from a computer .You will be able to turn on and off the

connected devices and read the sensor data from the interface and upload it to cloud.

● COMPONENTS NEEDED -

- **Raspberry PI 3**
- **PIR(Passive Infra Red) Sensor - for detecting human presence**
- **DHT 11 Sensor - for temperature and humidity**
- **BULB**
- **MOTOR or FAN**
- **Relay**
- **Power Source**
- **Buzzer**
- **Fan**
- **MQ5 Sensor - for detecting smoke**
- **Jumper wires**
- **LCD**

FLOW CHART-


FLOW DIAGRAM
PIR
DHT 11
MQ 5
DETECT INTRUDER
DETECT TEMPERATURE AND MOISTURE
DETECT SMOKE
RASPBERRY PI
INTRUDER PRESENT/ABSENT
LCD
UPLOAD TEMPERATURE AND MOISTURE
ON/OFF
ON/OFF
THINGS SPEAK
BUZZER
LED /BUZZER
THINGSPEAK
THINGSPEAK