*CONTENTS*

1. Abstract

2. Introduction

3. Modules Used

4. Working of a program

5. Detailed Explanation

6. Code Implementation

7. Program Flowchart

8. Importance

11.Applications

12. Conclusion

Battery Monitoring Python Program

**Abstract:**

This project provides a Python-based solution for monitoring the battery percentage of a computer. It offers proactive voice alerts when the battery level drops below a safe threshold and the device is not charging. The system is designed to assist users who may not always notice battery warnings visually, ensuring better, battery management and reducing the risk of sudden shutdowns.

**Introduction:**

This Python script is designed to monitor the battery percentage of your device and alert you with a voice message if the battery drops below a certain threshold and the charger is not connected.

**Modules Used:**

1. psutil – Used to get system hardware information such as battery status.

2. pyttsx3 – A text-to-speech conversion library in Python used for voice alerts.

3. time – Used to delay repeated checks (optional).

**Code Implementation:**

import psutil  
import pyttsx3  
import time  
  
def speak(text):  
 engine = pyttsx3.init()  
 engine.say(text)  
 engine.runAndWait()  
  
def check\_battery():  
 battery = psutil.sensors\_battery()  
 percent = battery.percent  
 plugged = battery.power\_plugged  
   
 print(f"Battery: {percent}% - Charging: {plugged}")  
   
 if percent < 10 and not plugged:  
 speak(f"Praneeth, your battery is less than 10%. Please charge your PC.")  
  
check\_battery()

**Program for background running automatically**

import time

import subprocess

while True:

# Run the target Python script

subprocess.Popen(["python", "battery\_percentage2.py"])

time.sleep(10)

**Working of the Program:**

1. The `check\_battery()` function uses `psutil.sensors\_battery()` to get the battery percentage and whether the device is plugged in.
2. If the battery level is below 10% and the device is not charging, a warning message is printed and also spoken aloud using the `speak()` function.
3. You can optionally run this check continuously every 5 minutes by uncommenting the `while True` loop.

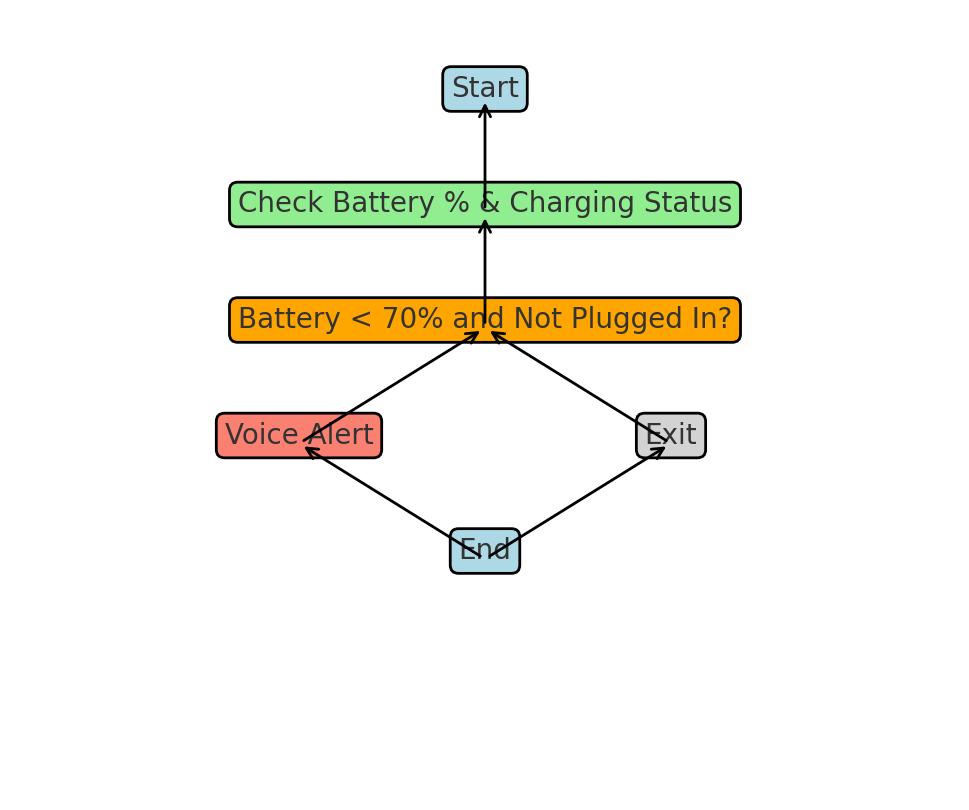
# Detailed Explanation:

This Python project is designed to help users get proactive battery alerts via voice notification.

The script uses the `psutil` module to monitor battery status and the `pyttsx3` library to produce voice output.  
When the battery percentage falls below a defined threshold (10% in this case) and the system is not plugged in a warning is spoken aloud. This functionality helps users stay informed without constantly monitoring the screen.  
The logic can be enhanced further by running the script continuously in the background or packaging it as a system tray utility.

Program Flowchart

The diagram below illustrates the logical flow of the battery monitoring system:

Advantages

1. Easy to use and implement, even for beginners.  
2. Provides real-time spoken battery alerts.  
3. Can prevent unexpected shutdowns by notifying in advance.  
4. Platform-independent and works offline.  
5. Reduces risk of unsaved work due to low battery.6. Alerts even inactive of program & background service support.

**Importance:**

* **Prevents Data Loss:**
  + Ensures the user is warned before battery drains completely.
* **Improves Device Care:**
  + Encourages timely charging and better battery maintenance.
* **Offline Functionality:**
  + Doesn't depend on internet (unlike some modern assistants).
* **Customizable & Extendable:**
  + Easily enhanced with GUI, logging, or periodic checking via schedulers or loops.

Applications

1. **Personal Battery Assistant**:
   * Helps users manage laptop power and avoid unexpected shutdowns.
2. **Productivity Tools**:
   * Prevents work loss by alerting users early about low battery.
3. : **Use Accessibility:**
   * Assists visually impaired users with voice-based battery alerts.
4. **Smart Home Integration** (with enhancements):
   * Can be part of a smart assistant setup for device health monitoring.
5. **Educational Projects**:

Good beginner Python project to learn about system resources and TTS.

**Conclusion**

This script is a simple yet effective way to get notified before your system battery runs out. It can be especially helpful when working without direct visibility of the battery icon.