**Design Report**

**Laptop Battery Status Monitor**

**– Under Guidance of Prof. G. Thakur**

**Presented By: -**

|  |  |  |
| --- | --- | --- |
| **Roll** | **SRN ID** | **Name** |
| 36 | 202101635 | Armaan Naik |

**Created & Presented: Python GUI Based & Designed with Adobe Photoshop**

**1. Project Overview:**

**1.1 Project Name:**

**Battery Status Monitor**

**1.2 Objective:**

**Develop a graphical interface application for real-time monitoring of a laptop's battery status.**

**Integrate the psutil library to access and retrieve accurate battery information.**

**Enhance user experience through visualizations of battery charge percentage and estimated time.**

**1.3 Rationale:**

**Users often require a convenient tool to monitor their laptop's battery status, aiding in decision-making related to charging and usage.**

**1.4 Scope:**

**Aimed at laptop users seeking a reliable solution for monitoring battery health.**

**Provides valuable insights into the current charging status, remaining charge, and estimated time for charging or discharging.**

**2. Code Structure:**

**2.1 Battery Information Functions:**

**update\_battery\_info():**

**Retrieves and processes battery information using the psutil library.**

**Determines the laptop's current charging state and adjusts the display accordingly.**

**Calculates and displays the estimated time to full charge or time left for discharge.**

**Utilizes label widgets to convey the current status, charge percentage, and estimated time.**

**Implements a canvas widget for a graphical representation of the battery charge.**

**refresh\_battery\_info():**

**Manually triggers the update of battery information, allowing users to refresh information on demand.**

**Graphical User Interface (Tkinter):**

**Configures the main window with an appropriate title and dimensions.**

**Integrates the PIL library for handling and resizing the banner image.**

**GUI elements include labeled widgets for status, charge percentage, and estimated time.**

**Utilizes a canvas widget for graphical representation, enhancing user comprehension.**

**Incorporates a manual refresh button (Refresh) for immediate updates.**

**3. Code Implementation:**

import tkinter as tk #standard Python interface-GUI components banane ke liye

from tkinter import ttk #widgets k enhance krne k liye (modern and consistent look) #ttk=Themed Tkinter

import psutil #battery status/info #Accessing system & process-related information.

from PIL import Image, ImageTk

#function to update battery information and visualization

def update\_battery\_info(): #function responsible- battery status update karne k liye & to display in GUI

    battery = psutil.sensors\_battery() #battery status & assigns it to battery variable

    charge\_percent = battery.percent #% & assigns it to c\_p variable

 #laptop charging pe hai ya nahi check karega

    if battery.power\_plugged: #examining the power\_plugged attribute of the battery object.

        status\_label.config(text="Charging")  #If the laptop is charging, this line updates a label widget called status\_label to display "Charging."

        # Calculate the estimated time to full charge

        if battery.secsleft == -1: #check if the est time to full charge is not avlble

            estimated\_time\_label.config(text="Charging (Calculating...)")

        else:

            # Calculate remaining time to fully charge the battery #based on current charge percentage and charging rate

            charging\_rate = battery.power\_plugged  # Charging rate=percentage of charge added per minute.

            remaining\_capacity = 100 - charge\_percent  # Remaining capacity to charge from current percentage to 100%

            remaining\_time\_minutes = remaining\_capacity / charging\_rate  # Time in minutes 39/1min

            hours = int(remaining\_time\_minutes // 60) #Floor/ciel division

            minutes = int(remaining\_time\_minutes % 60) #mod=remainder

            estimated\_time\_label.config(text=f"Time to Full: {hours} hours {minutes} minutes")

    else:

        status\_label.config(text="Discharging") #If the laptop is not on charging, this line updates a label widget called status\_label to display "Discharging." as already done ↑

        time\_left = battery.secsleft / 60 # calculates the time left to fully discharge the battery convert it into minutes.

        # Convert time left from minutes to hours and minutes

        hours = int(time\_left // 60) #Floor/ciel division

        minutes = int(time\_left % 60)  #mod=remainder

        estimated\_time\_label.config(text=f"Time Left: {hours} hours {minutes} minutes")

    charge\_label.config(text=f"Charge: {charge\_percent}%") #current charge % display karne ke liye

   #clear karne ke liye jo previous battery visualization the

    battery\_canvas.delete("all")

    #draw the battery outline

    battery\_canvas.create\_rectangle(10, 10, 100, 40, outline="black", width=2) #top-left corner-x,y & bottom-right corner-x,y

    #calculate the width of the battery based on the charge percentage (remining)

    battery\_width = (charge\_percent / 100) \* 80 # It is a proportion of the total width (80 pixels)

    #draw the filled battery based on charge percentage

    battery\_canvas.create\_rectangle(10, 10, 10 + battery\_width, 40, fill="green", width=0)

def refresh\_battery\_info():

    update\_battery\_info()

#main window banane ke liye

window = tk.Tk()

window.title("Battery Status Monitor")

window.geometry("800x500")

image = Image.open("batterysm.png")

image = image.resize((800, 160))

photo = ImageTk.PhotoImage(image)

image\_label = tk.Label(window, image=photo)

image\_label.pack(pady=10)

#labels banaya battery information ke liye=label widget for displaying the battery status.

status\_label = ttk.Label(window, text="", font=("JuliaMono", 16)) #status #referred-https://realpython.com/coding-font/

status\_label.pack(pady=10)

charge\_label = ttk.Label(window, text="", font=("JuliaMono", 16))  #charge percentage #referred-https://realpython.com/coding-font/

charge\_label.pack(pady=5)

estimated\_time\_label = ttk.Label(window, text="", font=("Arial", 16)) #est. time

estimated\_time\_label.pack(pady=5)

#widget banaya battery visualization k liye as told by ma'am

battery\_canvas = tk.Canvas(window, width=120, height=50)

battery\_canvas.pack(pady=5)

# update battery info button

refresh\_button = ttk.Button(window, text="Refresh", command=refresh\_battery\_info)

refresh\_button.pack(pady=10)

#Update battery info. n' visualization

update\_battery\_info()

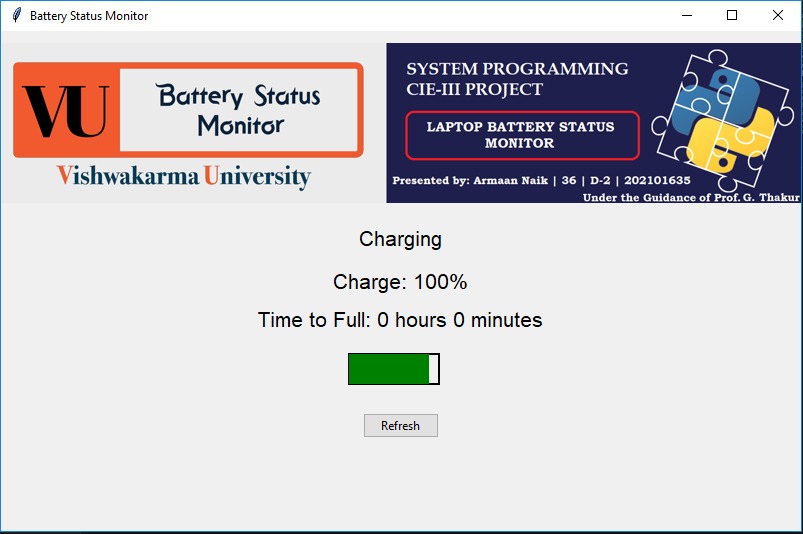
# #Automatic Update battery info./details n' visualization every 15 Sec. (60,000 milliseconds)

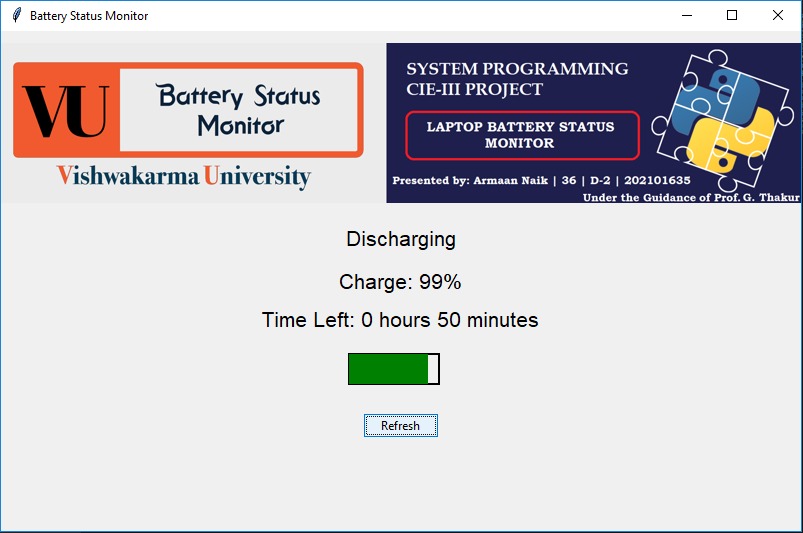
# window.after(1500, update\_battery\_info) #

#Starting GUI main loop

window.mainloop()

**4. Output Screenshot:**





**5. Conclusion:**

**5.1 User Experience:**

**The GUI offers an intuitive and informative display of essential battery metrics.**

**Visualizations contribute to a user-friendly interface, aiding in quick comprehension of the battery status.**

**5.2 Functionality:**

**Effectively utilizes the psutil library to access and process real-time battery information.**

**Visual elements, such as the graphical representation of the battery, enhance the overall user experience.**

**5.3 Automation:**

**The manual refresh button (Refresh) provides users with the flexibility to update battery information at their convenience.**

**Automatic updates every 15 seconds ensure continuous monitoring without user intervention.**

**5.4 Potential for Expansion:**

**The project lays the groundwork for future expansions, such as incorporating notifications for critical battery levels or maintaining a historical log of battery performance.**

**In summary, the Battery Status Monitor project provides a practical solution for users seeking an efficient and user-friendly tool for monitoring their laptop's battery health. The integration of real-time information and visualizations establishes a solid foundation for potential enhancements and extended functionalities in the realm of battery monitoring.**