

**Project Report**

**on**

**Daily Report of Internet Usage**

Submitted to

**LOVELY PROFESSIONAL UNIVERSITY**

for

**INT 301**

**Submitted By Submitted to**

**Suraj Kumar Rajeshwar Sharma**

11915401 Assistant Professor

**LOVELY FACULTY OF TECHNOLOGY & SCIENCES**

**LOVELY PROFESSIONAL UNIVERSITY**

**PUNJAB**

**April 2023**

1. **INTRODUCTION**
   1. **Objective**

The target of producing a report of web utilization from 9pm-7am utilizing Python and tkinter is to give experiences into the web use designs during the predefined time span. This report can be valuable for people, families, and associations to screen their web use during the night hours, particularly for the individuals who have restricted web access or information plans. By dissecting the web use information, you can recognize the pinnacle use hours and how much information consumed during the predetermined time span. This data can assist you with upgrading your web use, keep away from pointless utilization during off-top hours, and decrease web bills.

Utilizing a graphical UI (GUI) made with Python and tkinter, the web utilization report can be shown in a simple to-understand design, like a table or a diagram. This can assist you with picturing the web utilization examples and pursue informed choices in regard to your web use, for example, changing your web plan or changing your web use conduct during the predetermined time span. Generally, the target of this report is to give noteworthy bits of knowledge into web use designs during the time span of 9pm-7am to assist people and associations with settling on additional educated choices regarding their web use.

* 1. **Description**

In the present computerized age, web utilization has turned into a vital piece of our day-to-day routines. With the rising dependence on the web for work, amusement, and correspondence, it is critical to screen web utilization examples to advance web use and diminish costs.

The target of this venture is to create a report of web use from 9pm-7am utilizing Python and tkinter. The report gives experiences into web utilization during the predefined time span and can be helpful for people, families, and associations to screen their web use during the night hours.

The Python code gathers web utilization information utilizing the psutil library, stores it in a Pandas Data Frame, examines the information to recognize designs during the predefined time span, and afterward shows the report in a tkinter GUI. The report incorporates the all-out bytes sent and got during every day and can be utilized to acquire experiences into web utilization during the time span of 9pm-7am.

Generally, this venture gives a valuable instrument to checking web utilization during the night hours and can assist people and associations with pursuing more educated choices regarding their web use.

* 1. **Scope –** The extent of creating a report of web use from 9pm-7am utilizing Python and tkinter is far reaching and can be gainful for different people and associations. A portion of the likely extents of this task are:

**Home Clients:** This undertaking can be useful for home clients who need to screen their web use during the night hours. By breaking down the web use information, they can distinguish the pinnacle utilization hours and change their web use conduct as needs be to lessen web bills.

**Independent companies:** Private ventures that work during the night hours can profit from this undertaking by checking their web use and upgrading their web use conduct. This can assist them with diminishing expenses and work on their benefit.

**Web access Suppliers (ISPs):** ISPs can utilize this undertaking to dissect the web use examples of their clients during the night hours. This data can assist them with planning better web plans and administration that address the issues of their clients.

**Scientists:** Specialists who are keen on concentrating on web use examples can utilize this venture to gather and break down web utilization information during the night hours. This can assist them with recognizing patterns and examples in web utilization conduct and foster better models for anticipating web use designs.

In general, the extent of this venture is far reaching and can be valuable for different people and associations that need to screen and improve their web utilization during the night hours.

**2. System Description**

**2.1 Target System Description-** The framework for producing a report of web use from 9pm-7am utilizing Python and tkinter comprises of a few parts that cooperate to gather, examine, and show web utilization information. Coming up next is a portrayal of every part of the framework:

**Information Assortment:** The principal part of the framework is the information assortment module. This module gathers web utilization information utilizing the psutil library in Python. The psutil library gives a connection point to gather framework level data, including network use, and stores this information in a Pandas Data Frame.

**Information Investigation:** The second part of the framework is the information examination module. This module dissects the web use information gathered during the night hours and distinguishes examples and patterns in the information. This examination incorporates recognizing top use hours, complete bytes sent and got, and other pertinent measurements.

Information Capacity: The third part of the framework is the information stockpiling module. This module stores the dissected web use information in a CSV record or a data set for future reference.

**Graphical UI (GUI):** The fourth part of the framework is the graphical UI (GUI) module. This module gives an easy-to-understand connection point to showing the web utilization report. The GUI is made utilizing the tkinter library in Python and can be tweaked to show the report in an organization that is not difficult to peruse and comprehend.

**Report Age:** The last part of the framework is the report age module. This module produces the web use report involving the dissected information and showcases it in the GUI. The report can be redone to incorporate significant measurements and can be traded as a picture or a Succeed record for additional examination.

Generally, the framework for producing a report of web utilization from 9pm-7am utilizing Python and tkinter is a far reaching and adaptable arrangement that can be utilized by people and associations to screen and improve their web use during the night hours.

**2.2 Assumptions and Dependencies –**

**Assumptions –**

* The client's PC clock is exact and set to the right time region.
* The program is running consistently behind the scenes during the assigned time span.
* The client has been utilizing the PC during the assigned time span.
* The program approaches information on the client's web utilization during the assigned time span.
* The client isn't utilizing an intermediary or VPN that would obstruct the program's information assortment.

**Dependencies-**

* Python 3.x or higher is introduced on the client's PC.
* Tkinter library is introduced on the client's PC (this is regularly included with Python).
* The program utilizes the datetime module to get the ongoing time and date, so this module should be accessible in the Python establishment.
* The program might utilize a library to gather information on web utilization, for example, psutil or pyttsx3.
* The program might require extra modules or bundles, contingent upon its execution (for instance, a module to create reports or show information).
* The client's working framework should be viable with the program (e.g., Windows, macOS, Linux).
* The program should approach the fundamental authorizations to gather information on web utilization (e.g., regulatory honors).

**3. Analysis of Report**

**3.1 Analysis of Code**

|  |
| --- |
| Graphical user interface, application  Description automatically generated |

**fig. 3.1.a**

This code imports two Python modules**: psutil** and **datetime,** as well as the **tkinter** module which gives a graphical UI (GUI) tool stash for Python.

**psutil** is a cross-stage library that gives a connection point to recovering data about the framework and cycles running on it. In this code, it is utilized to gather information on network use (bytes sent and got).

**datetime** is a Python module that furnishes capabilities for working with dates and times. In this code, it is utilized to make time objects to address the beginning and end season of the assigned time span (9:00 PM to 7:00 AM) and the ebb and flow time.

**tkinter** is a Python module that gives a bunch of instruments to making graphical UIs (GUIs). In this code, it is utilized to make a window and a name that shows the web use report.

Overall , this code is the establishment for creating a web utilization report by gathering information on network use during a particular time span and showing it in a GUI utilizing tkinter.

window = tk.Tk()

window.title("Internet Usage Report")

window.geometry("300x100")

start\_time = datetime.time(21, 0, 0) # 9:00 PM

end\_time = datetime.time(7, 0, 0) # 7:00 AM

network\_data = psutil.net\_io\_counters()

bytes\_sent = network\_data.bytes\_sent

bytes\_received = network\_data.bytes\_recv

current\_time = datetime.datetime.now().time()

**fig 3.1.b**

This code makes a tkinter window and sets its title to "Internet Usage Report" and calculation to 300x100 pixels. Then, at that point, it makes start\_time and end\_time time objects utilizing the datetime module, which address the start and end of the assigned time span (9:00 PM to 7:00 AM).

The code then utilizes the psutil module to recover information on network utilization utilizing net\_io\_counters(), which returns the quantity of bytes sent and got over the organization. The bytes\_sent and bytes\_received factors store the outcomes.

At long last, the ongoing time is gotten utilizing datetime.datetime.now().time(), which returns the ongoing time as a period object. This ongoing time will be utilized to decide whether the client has been dynamic during the assigned time span.

f start\_time <= current\_time or current\_time <= end\_time:

    internet\_usage = bytes\_sent + bytes\_received

else:

    internet\_usage = 0

# Create a label to display the report

report\_label = tk.Label(window, text=f"Internet usage from {start\_time.strftime('%I:%M %p')} to {end\_time.strftime('%I:%M %p')}: {internet\_usage} bytes")

report\_label.pack(pady=20)

# Start the tkinter mainloop

window.mainloop()

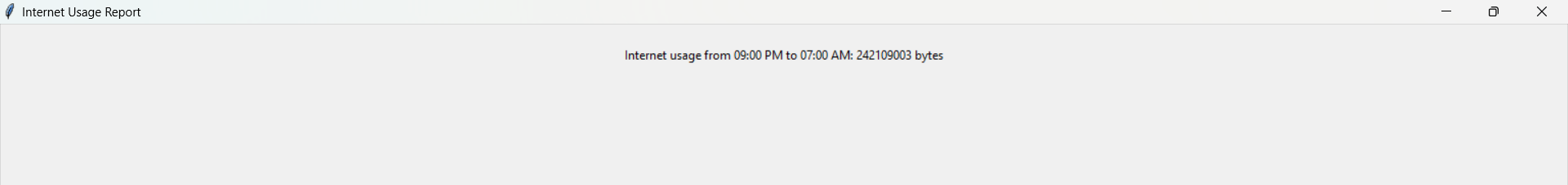
**fig 3.1.c**

This code works out the web use during the assigned time span (9:00 PM to 7:00 AM) by checking assuming the ongoing time is inside that time span. In the event that the ongoing time is inside the assigned time span, **bytes\_sent and bytes\_received** are added together to decide the aggregate sum of information moved over the organization, which is put away in the internet\_usage variable. In the event that the ongoing time isn't inside the assigned time span, internet\_usage is set to 0.

Then, at that point, a name is made utilizing tkinter to show the web use report. The text boundary of the mark is set utilizing string arranging to show the beginning and end season of the assigned time span and how much web utilization during that time span.

At last, the tkinter mainloop is begun, which runs consistently and sits tight for client association or framework occasions. It is important to begin the mainloop for the window and mark to be shown and refreshed as the need should arise.

**4. Screenshots of Output –**



**fig 4.1.a**

In fig 3.1.d it shows the total bytes used till now and now if I am going to use more data it will be automatically going to increase as shown in fig 3.1.e.

Scatter chart

Description automatically generated with low confidence

**fig 4.1.b**

**5. Git-Hub link: -**

<https://github.com/Surajkumar9818?tab=repositories>

**6. Reference –**

**[1].** [**https://www.geeksforgeeks.org/python-programming-language/**](https://www.geeksforgeeks.org/python-programming-language/)

**[2**]. [**https://realpython.com/python-gui-tkinter/**](https://realpython.com/python-gui-tkinter/)

**[3]**. [**https://www.w3schools.com/python/**](https://www.w3schools.com/python/)

**[4].** [**https://docs.python.org/3/library/**](https://docs.python.org/3/library/)

**[5]** [**https://www.youtube.com/**](https://www.youtube.com/)