Mini Project 1: [*Shutdown, Restart, and Logout using PC*]

1. Introduction

1.1 Problem Statement

Managing computer power operations like shutdown, restart, and logout typically requires navigating through menus or using manual commands. This can be time-consuming and inefficient for users who prefer quick execution.

1.2 Objectives

- *To develop a simple GUI-based application to quickly execute shutdown, restart, and logout commands.*
- *To enhance user productivity by providing one-click operations.*
- *To implement a cross-platform-friendly script (for Windows) using Python.*

1.3 Scope of the Project

The project will be limited to Windows-based systems and will use Python to execute system commands. It will be useful for personal use, lab environments, and automated system management.

2. Technology Stack Used

2.1 Programming Languages

- *Python*

2.2 Libraries/Frameworks

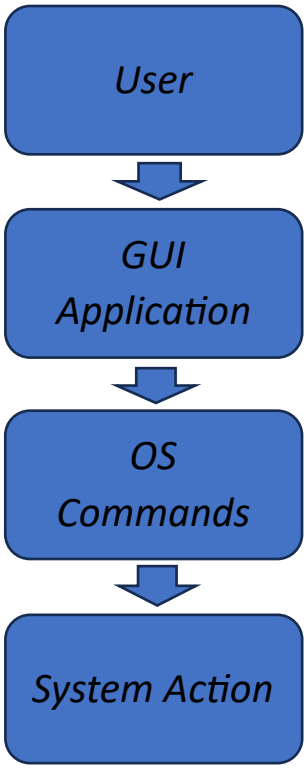
- *Tkinter (for GUI)*
- *OS module (for executing system commands)*

2.3 Tools and Platforms

- *Python IDLE / VS Code*
- *Windows 10/11 operating system*

3. System Architecture

3.1 Architecture Diagram



3.2 Module Description

- ***GUI Module:*** *Handles the interface with buttons for shutdown, restart, and logout.*
- ***Command Execution Module:*** *Executes respective system commands using Python's os.system() function.*

4. Dataset Description (if applicable)

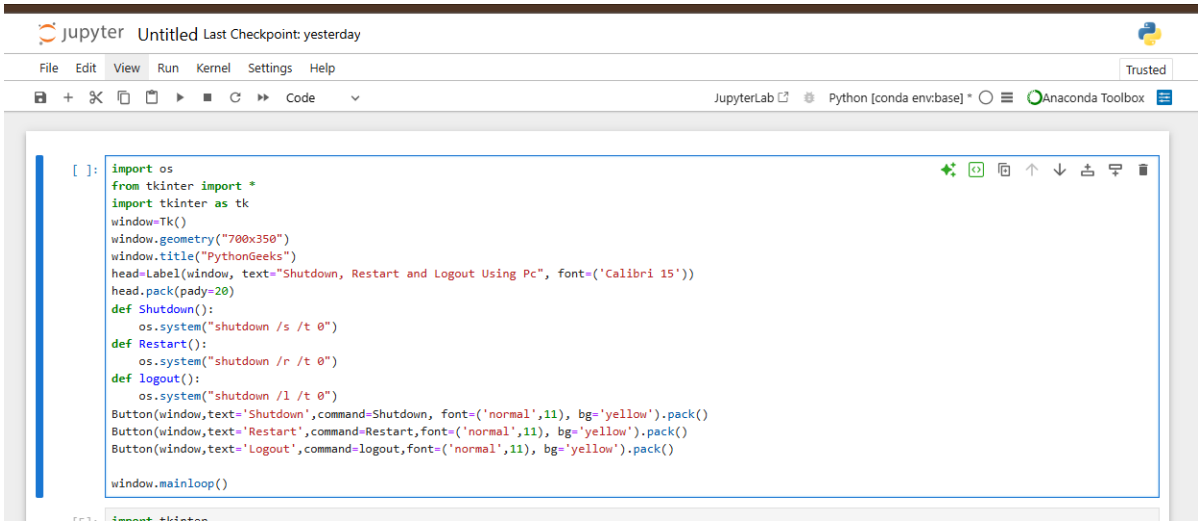
- *Not applicable for this project.*

5. Implementation

5.1 Code Flow Description

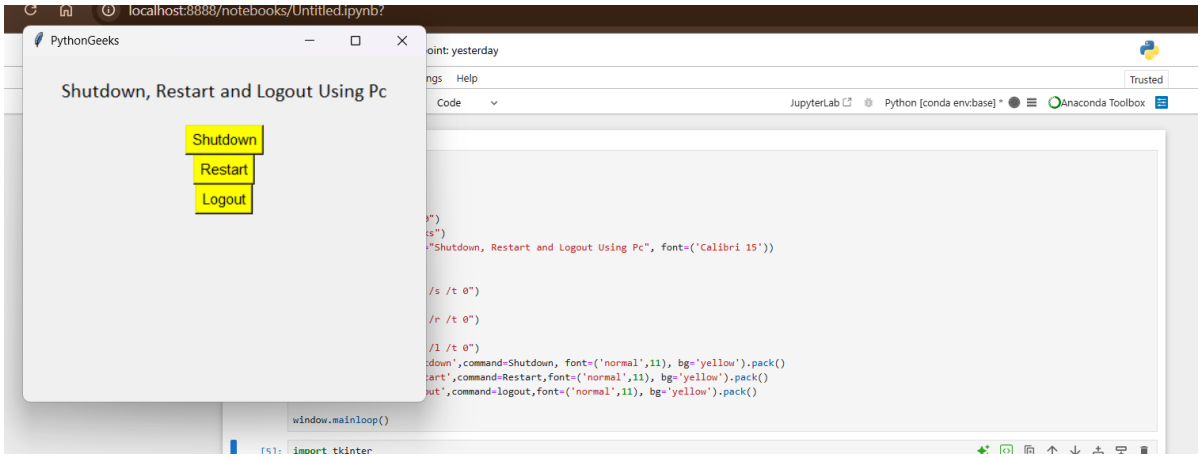
- *User opens the application.*
- *The main window is displayed with three buttons: Shutdown, Restart, Logout.*
- *When a button is clicked, the corresponding OS command is executed.*
- *The system performs the action immediately or after a small delay.*

5.2 Screenshots of Execution



6. Results and Analysis

6.1 Output Samples



6.2 Performance Evaluation

- *Execution Time: Instantaneous (<1 second command execution).*
- *Memory Usage: Minimal (<20 MB).*
- *User Feedback: Intuitive and fast.*

7. Challenges Faced and Solutions

- ***Challenge:** Commands differ across operating systems.*
- ***Solution:** Limited the scope to Windows and documented changes for Linux/Mac.*
- ***Challenge:** Preventing accidental clicks.*
- ***Solution:** Added confirmation pop-ups before executing actions.*

8. Conclusion

The project successfully automates system power operations via a simple Python GUI. It saves time and improves convenience for users who perform these actions frequently.

9. References

https://docs.python.org/3/library/tkinter.html
https://docs.python.org/3/library/random.html
https://realpython.com/python-password-generator/