Password Strength Checker — Project Report

Problem Statement

Weak passwords are among the most common causes of data breaches. Users often choose short, predictable, or dictionary-based passwords without knowing the risk. There's a need for a tool that can help users evaluate and improve their password strength instantly and interactively.

Objective

To design and develop a desktop GUI application using Python that allows users to:

- Input passwords securely
- Receive real-time feedback on password strength
- View color-coded strength indicators and progress meter
- Get actionable suggestions to improve weak passwords
- Toggle password visibility for usability

Tools and Technologies Used

Tool/Library	Purpose
Python 3	Core programming language
Tkinter	GUI development framework
Regex (re**)**	Pattern matching for strength rules
ttk	Themed widgets (like Progressbar)

System Architecture

Modular Design:

- gui.py: GUI and interaction logic
- checker.py: Password evaluation using regex
- utils.py: Improvement suggestions
- requirements.txt: Dependencies list

Features

- Real-time password strength evaluation as user types
- Progress bar meter with visual strength level
- Color-coded text: Red (Weak), Orange (Moderate), Green (Strong)
- Dynamic suggestions to improve weak passwords
- Show/Hide password toggle for better usability
- Modern, clean, responsive UI

Password Strength Criteria

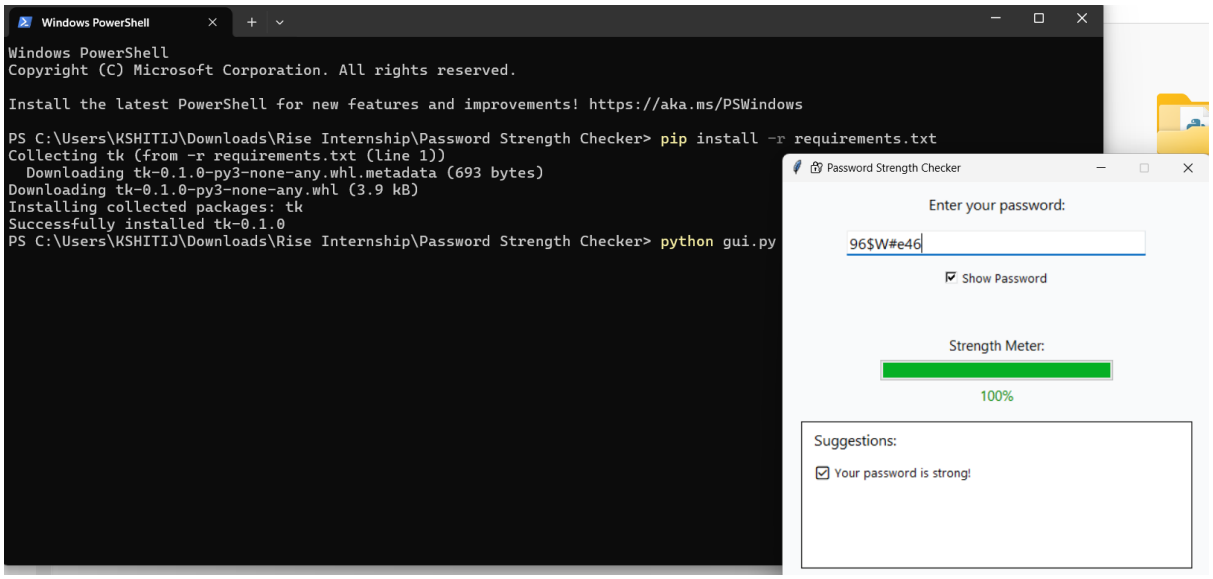
The following conditions are checked:

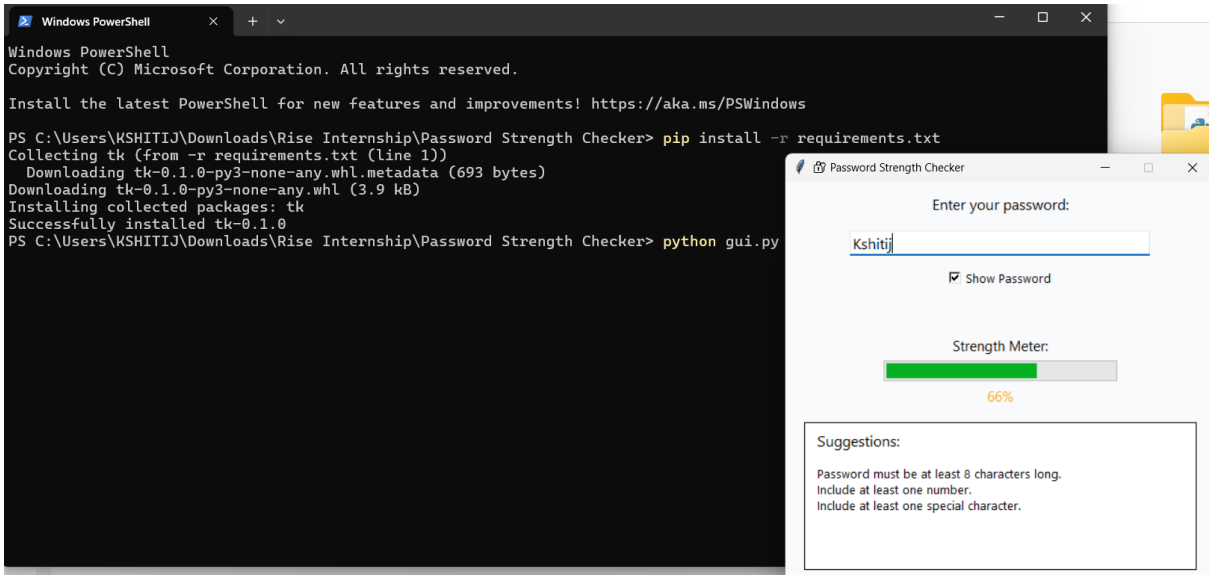
- Minimum **8 characters**
- At least **1 uppercase letter**
- At least **1 lowercase letter**
- At least **1 number**
- At least **1 special character** (@#$%^&*!?)

Passwords are rated as:

- **Weak**: Meets < 3 conditions
- **Moderate**: Meets 3–4 conditions
- **Strong**: Meets all 5 conditions

User Interface (Screenshots)





How to Run

Requirements:

- Python 3.x
- Tkinter (usually included)
- Optional: nltk (pip install nltk)

Steps:

1. Clone or unzip the project
2. Navigate to the project folder
3. Run: pip install -r requirements.txt
4. python gui.py

Future Enhancements

- Detect dictionary-based passwords using NLTK
- Encrypted password suggestions generator
- Password history with charts and reports
- Convert to .exe or standalone app
- Web version using Flask or Django

Conclusion

This project offers a functional, real-world solution for improving password security awareness. With a responsive GUI and modular code structure, it's ideal for learners, professionals, and recruiters seeking practical cybersecurity skills. The tool encourages strong password habits — a critical component of secure digital identity.