PROJECT-2 PASSWORD STRENGHT CHECK (GUI-BASED)

Problem Statement:

Weak and predictable passwords are one of the most common reasons for data breaches and unauthorized access. This project aims to help users choose stronger passwords by analyzing their input and giving real-time feedback on password strength.

Objective:

To develop a **Python GUI-based application** that:

- Analyzes the strength of a password entered by the user
- Provides improvement suggestions for weak passwords
- Offers additional features like random password generation and password visibility toggle.

Technologies & Tools Used:

Tool/Technology	**Purpose**
Python	Programming language used for logic and backend
Tkinter	GUI creation for desktop application
Regex (`re`)	Pattern matching (for checking letters, digits, symbols)
NLTK (optional)	To detect dictionary words (makes password weaker)
`random`, `string`	To generate strong random passwords
ttk.Progressbar	To visualize password strength level

How It Works:

First create a file named password_strenghtcheck.py and add the python code below.

```
   import tkinter as tk

from tkinter import ttk

import re

import random

import string

from nltk.corpus import words
```

```
import nltk


# Download word list (only once)

try:

    english_words = set(words.words())

except LookupError:

    nltk.download('words')

    english_words = set(words.words())


# Password strength logic

def check_password_strength(password):

    score = 0

    suggestions = []


    if len(password) >= 8:

        score += 1

    else:

        suggestions.append("➡ Make it at least 8 characters long.")


    if re.search(r"[A-Z]", password):

        score += 1

    else:

        suggestions.append("➡ Add at least one uppercase letter.")


    if re.search(r"[a-z]", password):
```

```
    score += 1
else:
    suggestions.append("➡ Add at least one lowercase letter.")

if re.search(r"[0-9]", password):
    score += 1
else:
    suggestions.append("➡ Include at least one digit.")

if re.search(r"[!@#$%^&*(),.?\":{}|<>]", password):
    score += 1
else:
    suggestions.append("➡ Use special characters like !, @, #, etc.")

if password.lower() in english_words:
    suggestions.append("➡ Avoid using dictionary words.")

# Determine strength text
if score >= 4 and not suggestions:
    return "✅ Strong Password", suggestions, 100
elif score >= 3:
    return "⚠ Moderate Password", suggestions, 60
else:
    return "❌ Weak Password", suggestions, 30
```

```
# Show/hide password

def toggle_password():

    if show_var.get():

        entry.config(show='')

    else:

        entry.config(show='*')


# Generate random strong password

def generate_password():

    chars = string.ascii_letters + string.digits + string.punctuation

    random_pw = ''.join(random.choice(chars) for _ in range(12))

    entry.delete(0, tk.END)

    entry.insert(0, random_pw)

    check()


# Check password strength

def check():

    password = entry.get()

    strength, suggestions, score = check_password_strength(password)

    result_label.config(text=strength, fg="green" if "Strong" in strength else "red")


    suggestion_label.config(text="\n".join(suggestions))

    strength_bar['value'] = score


# GUI setup
```

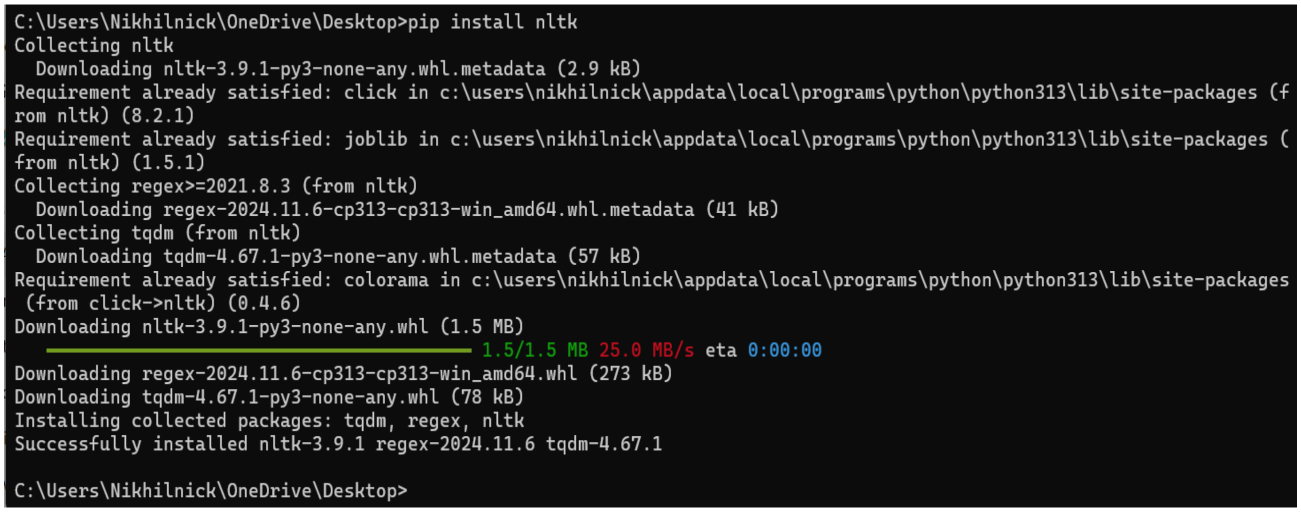
```
root = tk.Tk()

root.title("Password Strength Checker")

root.geometry("500x400")

root.resizable(False, False)


# Entry and labels

tk.Label(root, text="🔐 Enter your password:", font=('Arial', 12)).pack(pady=10)


entry = tk.Entry(root, width=35, show='*', font=('Arial', 12))

entry.pack()


show_var = tk.BooleanVar()

show_check = tk.Checkbutton(root, text="Show Password", variable=show_var, command=toggle_password)

show_check.pack()


tk.Button(root, text="Check Strength", command=check, font=('Arial', 12)).pack(pady=5)


tk.Button(root, text="Generate Strong Password", command=generate_password, font=('Arial',
12)).pack(pady=5)


result_label = tk.Label(root, text="", font=('Arial', 14, 'bold'))

result_label.pack(pady=10)


strength_bar = ttk.Progressbar(root, orient="horizontal", length=300, mode='determinate')

strength_bar.pack(pady=5)
```

suggestion_label = tk.Label(root, text="", font=('Arial', 10), fg="blue", wraplength=450, justify="left")

suggestion_label.pack()

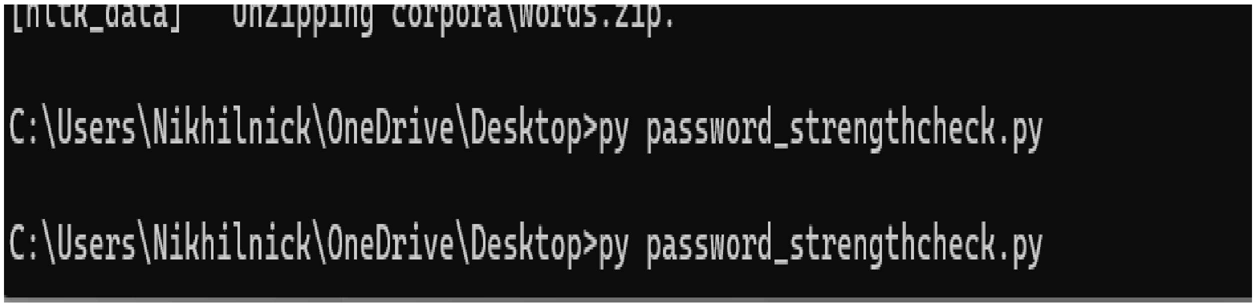
root.mainloop()

install nltk using command pip install nltk.



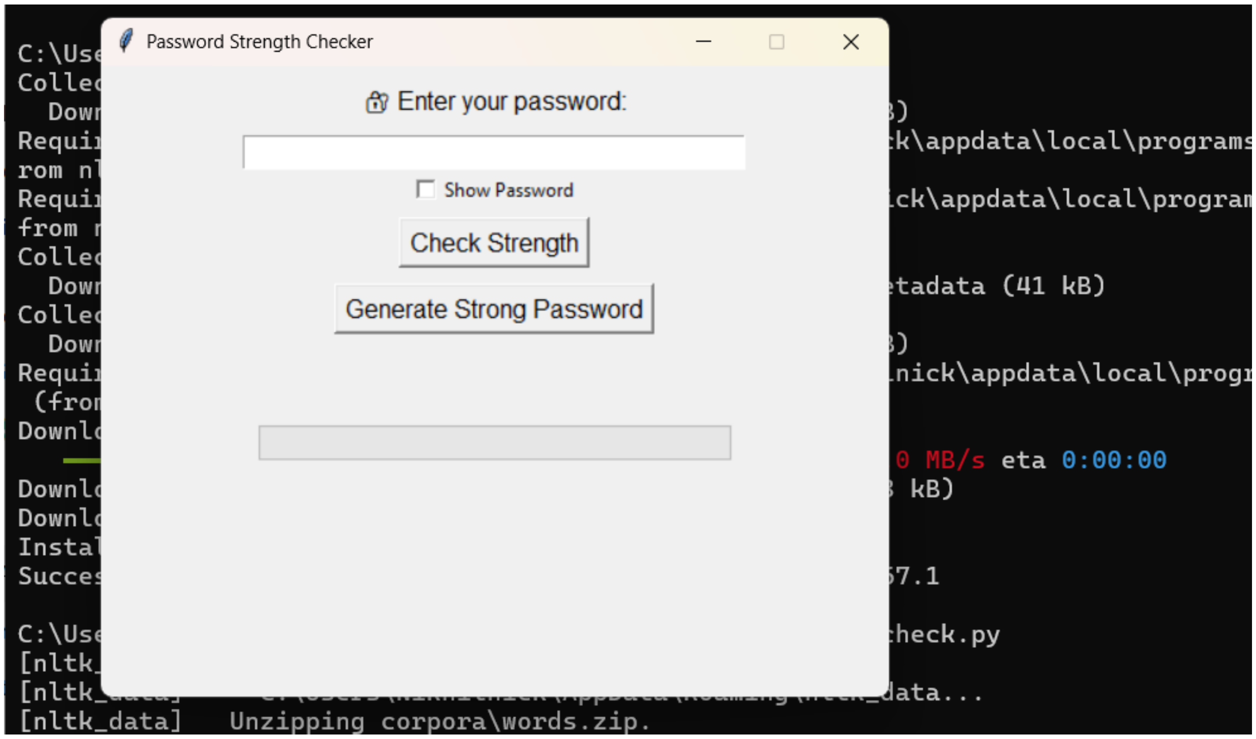
```
C:\Users\Nikhilnick\OneDrive\Desktop>pip install nltk
Collecting nltk
  Downloading nltk-3.9.1-py3-none-any.whl.metadata (2.9 kB)
Requirement already satisfied: click in c:\users\nikhilnick\appdata\local\programs\python\python313\lib\site-packages (f
rom nltk) (8.2.1)
Requirement already satisfied: joblib in c:\users\nikhilnick\appdata\local\programs\python\python313\lib\site-packages (
from nltk) (1.5.1)
Collecting regex>=2021.8.3 (from nltk)
  Downloading regex-2024.11.6-cp313-cp313-win_amd64.whl.metadata (41 kB)
Collecting tqdm (from nltk)
  Downloading tqdm-4.67.1-py3-none-any.whl.metadata (57 kB)
Requirement already satisfied: colorama in c:\users\nikhilnick\appdata\local\programs\python\python313\lib\site-packages
 (from click->nltk) (0.4.6)
Downloading nltk-3.9.1-py3-none-any.whl (1.5 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.5/1.5 MB 25.0 MB/s eta 0:00:00
Downloading regex-2024.11.6-cp313-cp313-win_amd64.whl (273 kB)
Downloading tqdm-4.67.1-py3-none-any.whl (78 kB)
Installing collected packages: tqdm, regex, nltk
Successfully installed nltk-3.9.1 regex-2024.11.6 tqdm-4.67.1

C:\Users\Nikhilnick\OneDrive\Desktop>
```

Run the code password_strengthcheck file.



```
[nltk_data]   Unzipping corpora\words.zip.

C:\Users\Nikhilnick\OneDrive\Desktop>py password_strengthcheck.py

C:\Users\Nikhilnick\OneDrive\Desktop>py password_strengthcheck.py
```

After running the above command we can see the graphical interface .



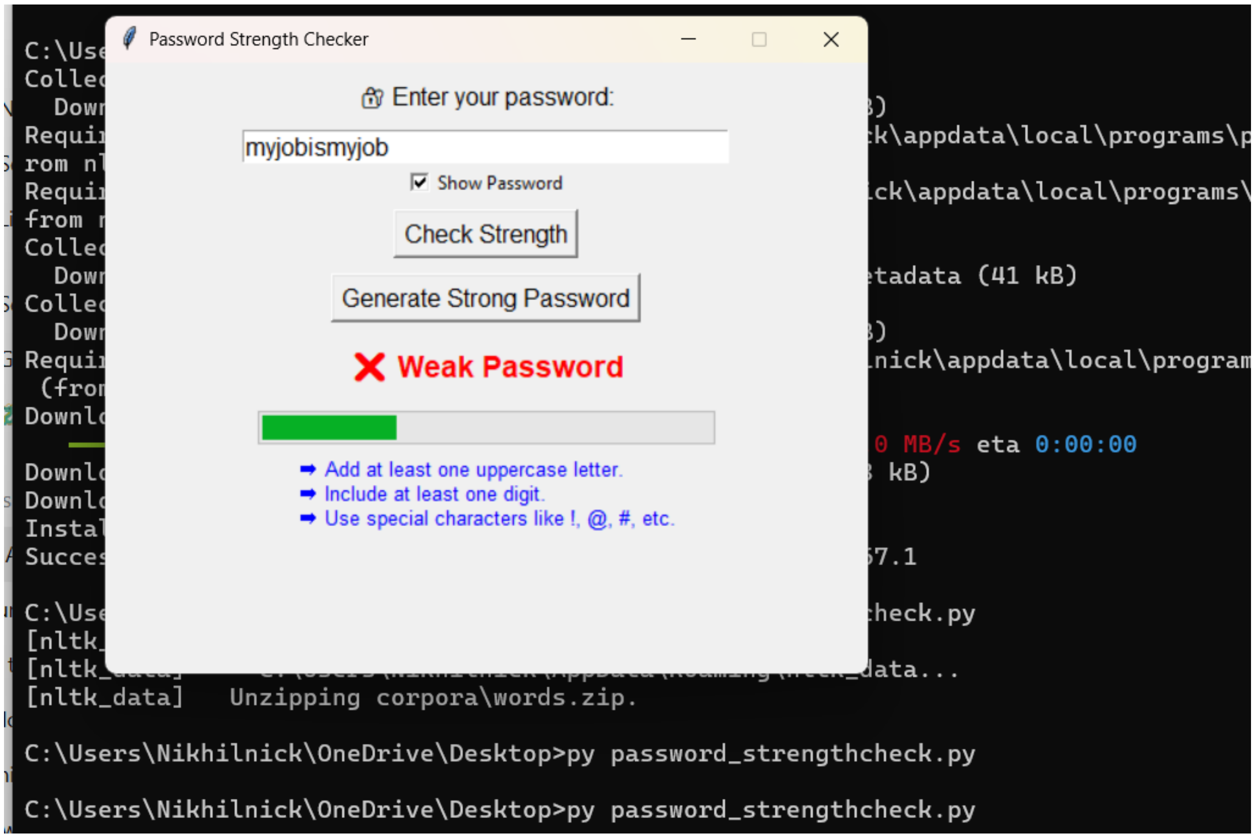
```
C:\Use
Colle
  Down                                                    3)
Requi                                                    k\appdata\local\programs
rom nl
Requi                                                    ck\appdata\local\program
from n
Colle
  Down                                                   tadata (41 kB)
Colle
  Down                                                    3)
Requi                                                    nick\appdata\local\progr
 (fron
Downl
   ━━                                                    0 MB/s eta 0:00:00
Downl                                                    3 kB)
Downl
Insta
Succe                                                    57.1

C:\Use                                                   heck.py
[nltk_
[nltk_                                                   data...
[nltk_data]    Unzipping corpora\words.zip.
```

Steps to check the password strength.

1. User enters a password into the input field.
2. Clicks **"Check Strength"**.
3. The app:
    - Evaluates the password using regex checks
    - Compares it against a dictionary word list (optional)
    - Gives a score (out of 5)
    - Displays a message like: Strong / Moderate / Weak
    - Shows a progress bar and improvement tips (if needed)
4. User can also click **"Generate Strong Password"** to get a random secure password.
5. "Show Password" toggle helps verify typed input.

Observation and Testing:

Testing some random passwords to know the strength of passwords, below are some screenshots.



Password Strength Checker

🔐 Enter your password:

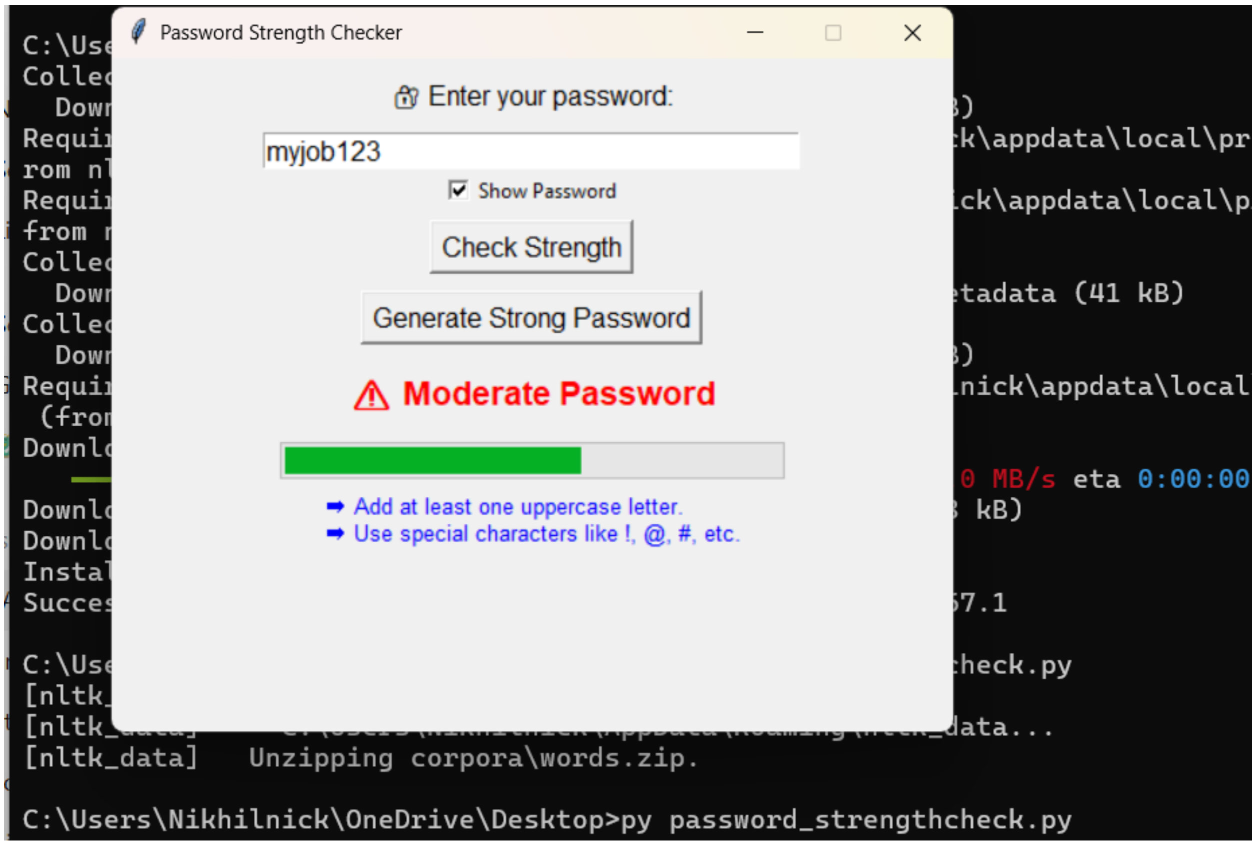
myjobismyjob

☑ Show Password

❌ Weak Password

➡ Add at least one uppercase letter.
➡ Include at least one digit.
➡ Use special characters like !, @, #, etc.

```
[nltk_data]    Unzipping corpora\words.zip.

C:\Users\Nikhilnick\OneDrive\Desktop>py password_strengthcheck.py

C:\Users\Nikhilnick\OneDrive\Desktop>py password_strengthcheck.py
```



Password Strength Checker

🔐 Enter your password:

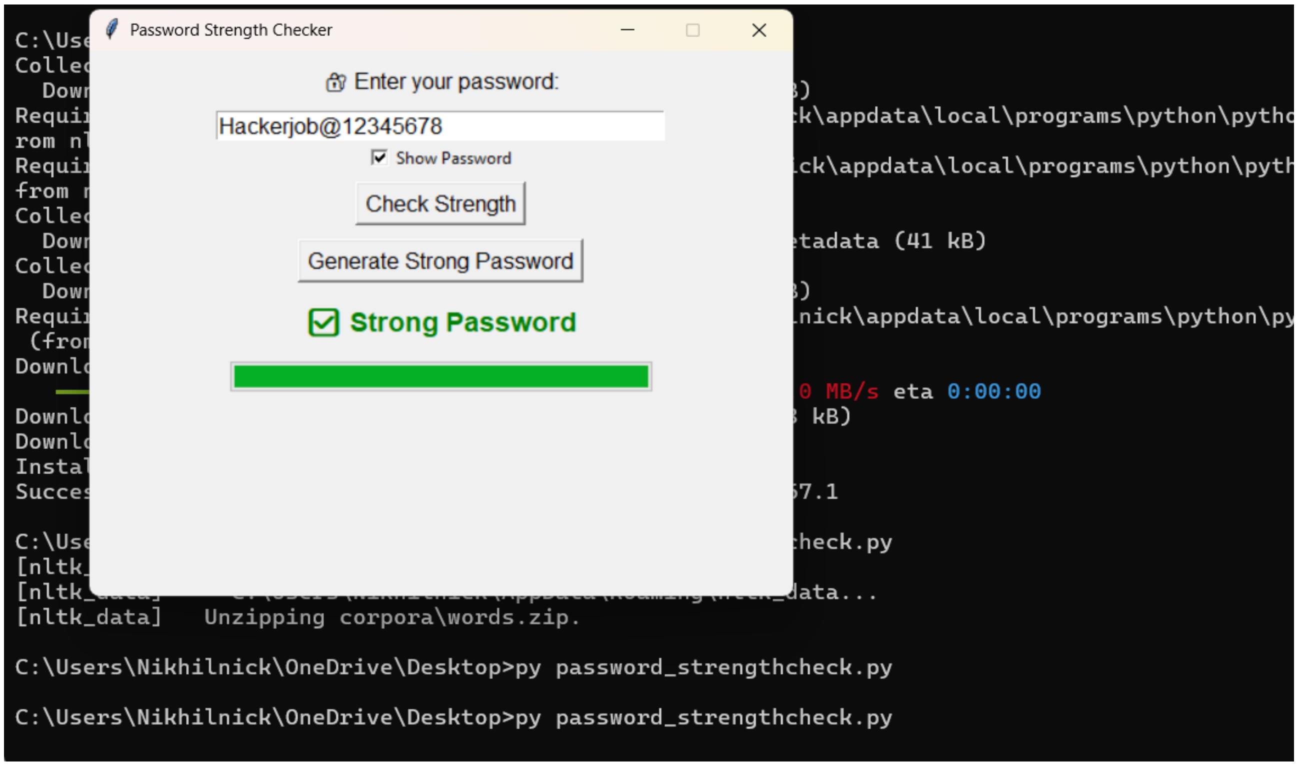
myjob123

☑ Show Password

⚠ Moderate Password

➡ Add at least one uppercase letter.
➡ Use special characters like !, @, #, etc.

```
[nltk_data]    Unzipping corpora\words.zip.

C:\Users\Nikhilnick\OneDrive\Desktop>py password_strengthcheck.py
```



Hackerjob@12345678

✓ Strong Password

```
C:\Use
Collec
  Down                                                          3)
Requi                                                           k\appdata\local\programs\python\pytho
rom nl
Requi                                                           ck\appdata\local\programs\python\pyth
from r
Collec
  Down                                                          tadata (41 kB)
Collec
  Down                                                          3)
Requi                                                           nick\appdata\local\programs\python\py
 (fro
Downl
                                                                0 MB/s eta 0:00:00
Downl                                                           3 kB)
Downl
Insta
Succes                                                          57.1

C:\Use                                                          heck.py
[nltk_
[nltk_                                                          data...
[nltk_data]   Unzipping corpora\words.zip.

C:\Users\Nikhilnick\OneDrive\Desktop>py password_strengthcheck.py

C:\Users\Nikhilnick\OneDrive\Desktop>py password_strengthcheck.py
```

Generating a Random password which is safe to use.

Key Features Implemented:

Features	Description
Password Strength Analyzer	Checks for length, uppercase, lowercase, digit, special characters
Real-time Feedback	Displays whether the password is Strong, Moderate, or Weak
Improvement Suggestions	Tells the user what is missing in their password
Progress Bar	Visual representation of password strength (30%/60%/100%)
Show/Hide Password Toggle	Users can choose to view the password while typing
Random Password Generator	Suggests some strong passwords .

Conclusion:

This project serves as a practical password auditing tool for everyday users. It not only checks password strength but also promotes good password hygiene practices through suggestions and automation features.