Elevate Labs Internship Project

Password Strength Analyzer with Custom Wordlist Generator

June 23, 2025

1 Introduction

This project is a Password Strength Analyzer with a Custom Wordlist Generator, designed to improve cybersecurity awareness. Passwords are critical for protecting online accounts, but weak passwords like "123" are easily hacked. This tool checks password strength and creates wordlists (lists of possible passwords) based on user inputs such as names or dates. These wordlists help understand how hackers guess passwords in ethical hacking tests. By building this tool, I learned about secure passwords and their role in preventing unauthorized access.

2 Abstract

The Password Strength Analyzer evaluates password security using the zxcvbn library and generates custom wordlists from user inputs like name, date, or pet name. It features a user-friendly tkinter graphical interface. Users can test passwords for strength (scored 0–4) and save wordlists as text files, aiding cybersecurity education and ethical hacking practice.

3 Tools Used

I used Python, a programming language, to build the tool. The zxcvbn-python library analyzes password strength, while tkinter creates the graphical interface. The itertools and re libraries generate wordlist variations, such as leetspeak (e.g., "hemanth2001" for "HEMANTH2001"). These tools are free, easy to install, and work on Windows, Mac, and Linux.

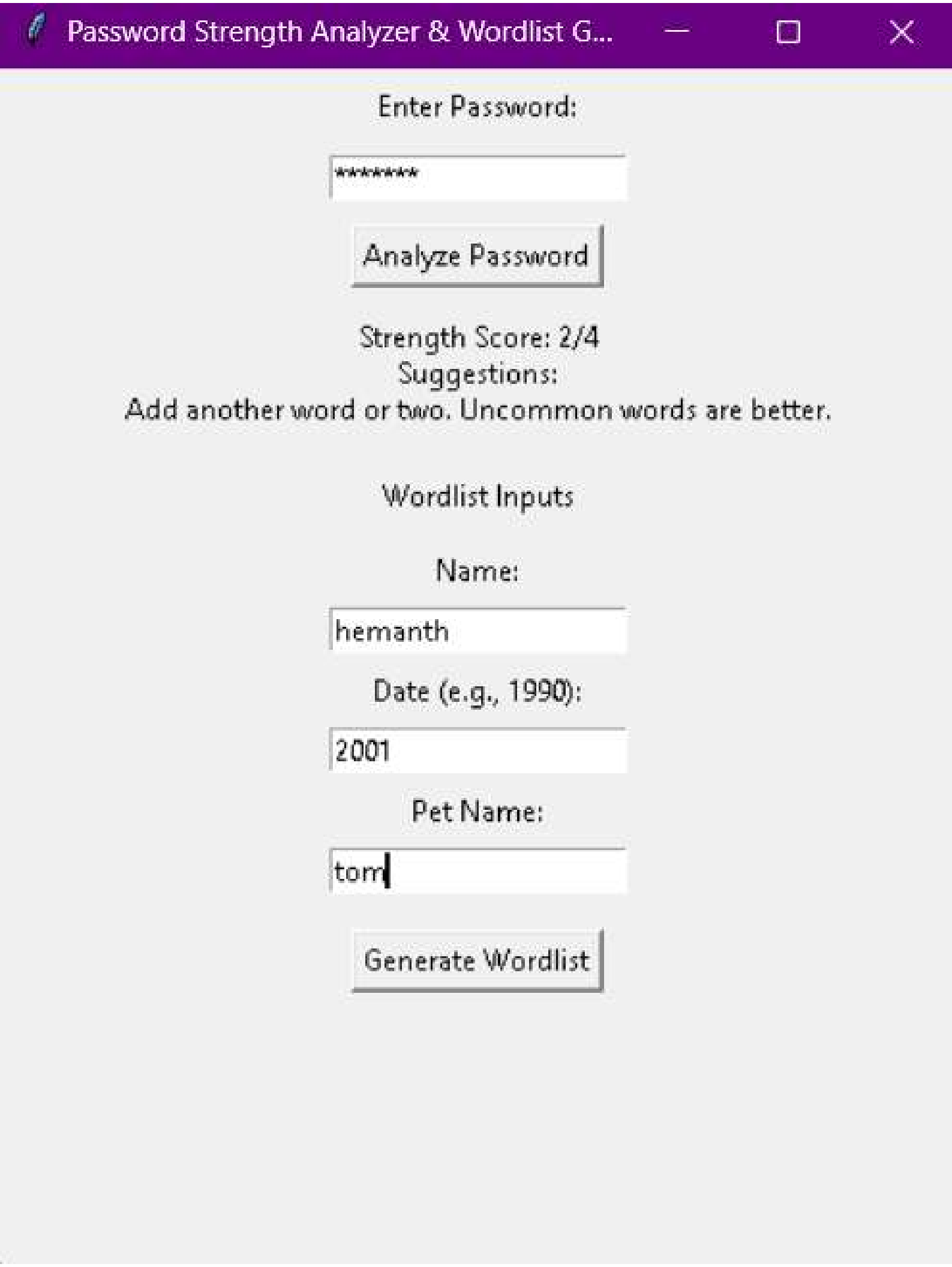
4 Steps Involved

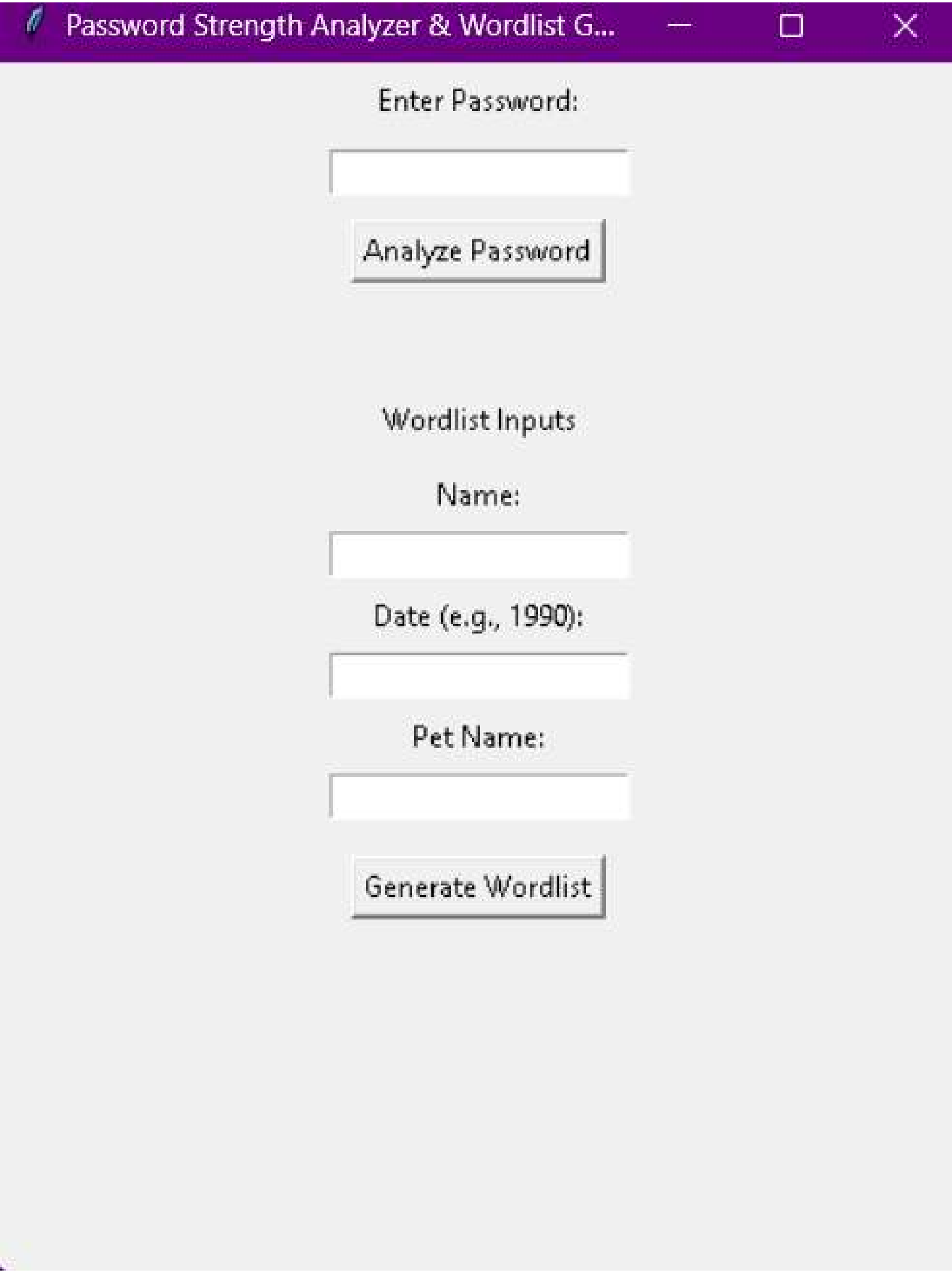
- Installed Python 3.8+ and the zxcvbn-python library using the command `pip install zxcvbn-python`.
- Wrote code to analyze passwords with zxcvbn, which assigns scores (0 for weak, 4 for strong) and offers suggestions like "Add uppercase letters."
- Developed a wordlist generator that creates variations from inputs (e.g., name "Tom," date "2001") like "HEMANTH2001", "hemanth2001", and "tom 2001".
- Built a tkinter GUI with fields for passwords and inputs, plus buttons to analyze and generate wordlists.
- Tested weak passwords (e.g., "123"), strong passwords (e.g., "P@ssw0rd2025!"), and wordlist generation with various inputs.
- Saved the wordlist as `wordlist.txt` and verified outputs (e.g., "tom 2001").

- Took screenshots for documentation.

5 Conclusion

The Password Strength Analyzer helps users create secure passwords and learn about wordlist-based hacking techniques. Its GUI is accessible for beginners. I gained knowledge of Python programming and password security. The tool can be enhanced by adding more wordlist patterns (e.g., beyond "HEMANTH2001" or "tom 2001") or cloud storage for results. Its a valuable resource for cybersecurity education and ethical hacking awareness.


Password Strength Analyzer & Wordlist G...
Enter Password:
****
Analyze Password
Strength Score: 2/4
Suggestions:
Add another word or two. Uncommon words are better.
Wordlist Inputs
Name:
hemanth
Date (e.g., 1990):
2001
Pet Name:
tom
Generate Wordlist


Password Strength Analyzer & Wordlist G...
Enter Password:
Analyze Password
Wordlist Inputs
Name:
Date (e.g., 1990):
Pet Name:
Generate Wordlist


Password Strength Analyzer & Wordlist G...
Enter Password:
****
Analyze Password
Strength Score: 2/4
Suggestions:
Add another word or two. Uncommon words are better.

VULNERA.txt | password_analyzer.py | wordlist.txt

File Edit View

```
2001
20011
20012001
HEMANTH
HEMANTH1
HEMANTH2001
Hemanth
TOM
TOM1
TOM2001
Tom
h3m@nth
h3manth
hem@nth
hemanth
hemanth1
hemanth2001
t0m
tom
tom1
tom2001
```