PASSWORD STRENGTH ANALYZER WITH CUSTOM WORDLIST GENERATOR

PROJECT REPORT

Submitted By:

Dyuthi D

ABSTRACT

This project focuses on developing a Python-based tool that analyzes the strength of user-provided passwords and generates custom wordlists for security testing.

It evaluates password strength using zxcvbn and creates dynamic wordlists based on user inputs such as name, date of birth, and pet name.

The tool can be used to educate users on password security and demonstrate how weak or predictable passwords can be easily guessed using custom-generated wordlists.

INTRODUCTION

Passwords are one of the most commonly used methods for authentication, yet many users still choose weak and predictable passwords. This project addresses the need for strong password practices by providing a tool that not only measures password strength but also highlights potential vulnerabilities through custom wordlists.

By generating user-specific wordlists, the project demonstrates how attackers could use personal information to compromise accounts. This enhances user awareness and promotes the creation of stronger, less predictable passwords.

DESIGN

Tools and Technologies

1. Python: The main programming language used.
2. zxcvbn: A library to evaluate password strength.
3. NLTK: Used for text processing in future expansions.
4. VS Code: For coding and debugging

IMPLEMENTATION

The project is divided into two main components:

1. Password Strength Analyzer: -

- Takes a user password as input.
- Uses **zxcvbn** to determine the password's strength score, crack time, and suggestions.

2. Custom Wordlist Generator: -

- Takes user inputs (name, pet name, date of birth, etc.).
- Generates a set of possible passwords by combining patterns (e.g., appending numbers, reversing names, leetspeak).
- Stores the generated words in a text file (wordlist.txt).
- The tool is menu-driven and easy to use from the command line.

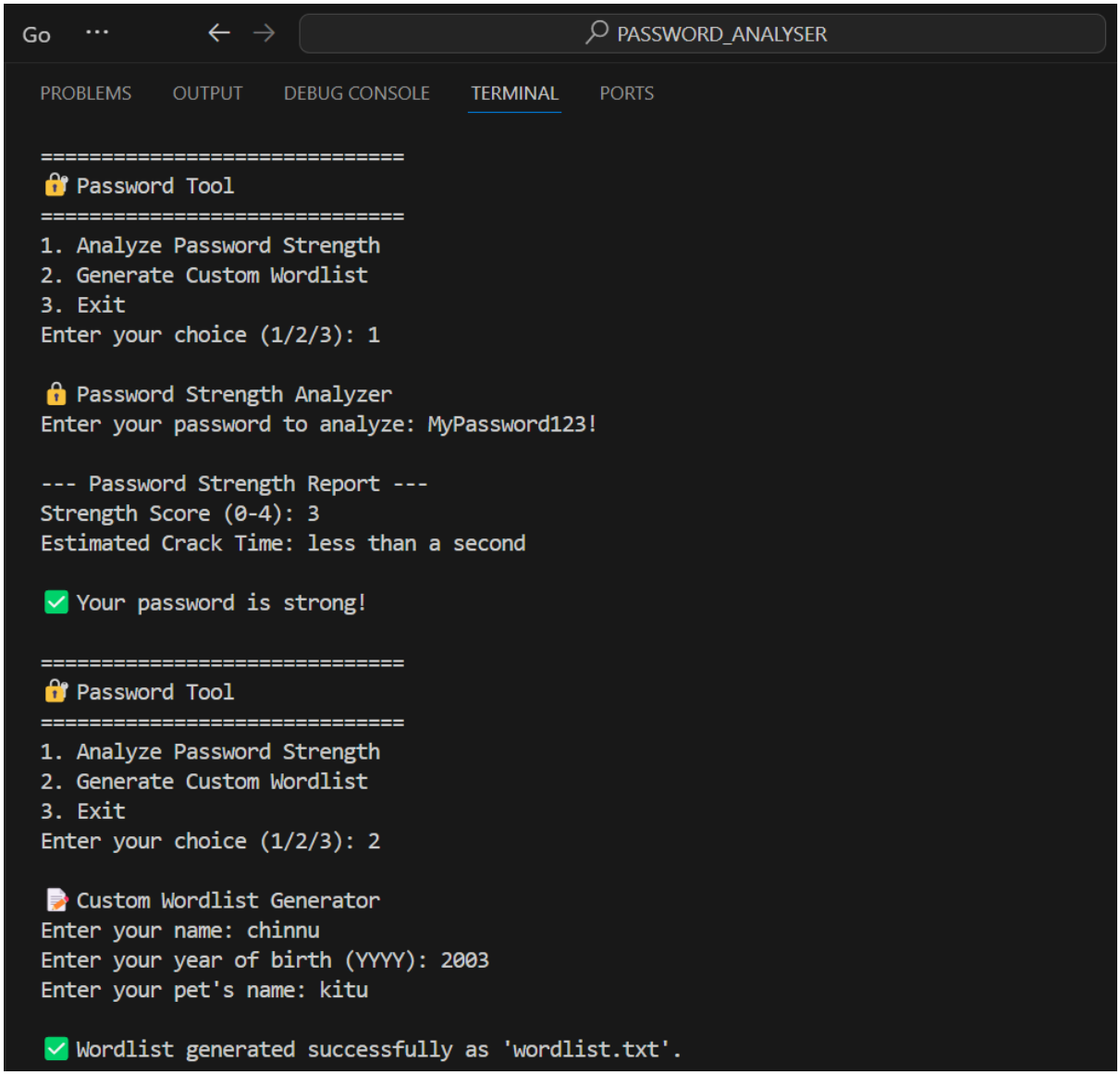
RESULTS

The final tool provides users with a clear understanding of their password strength and the ability to see how attackers could guess passwords using custom wordlists.

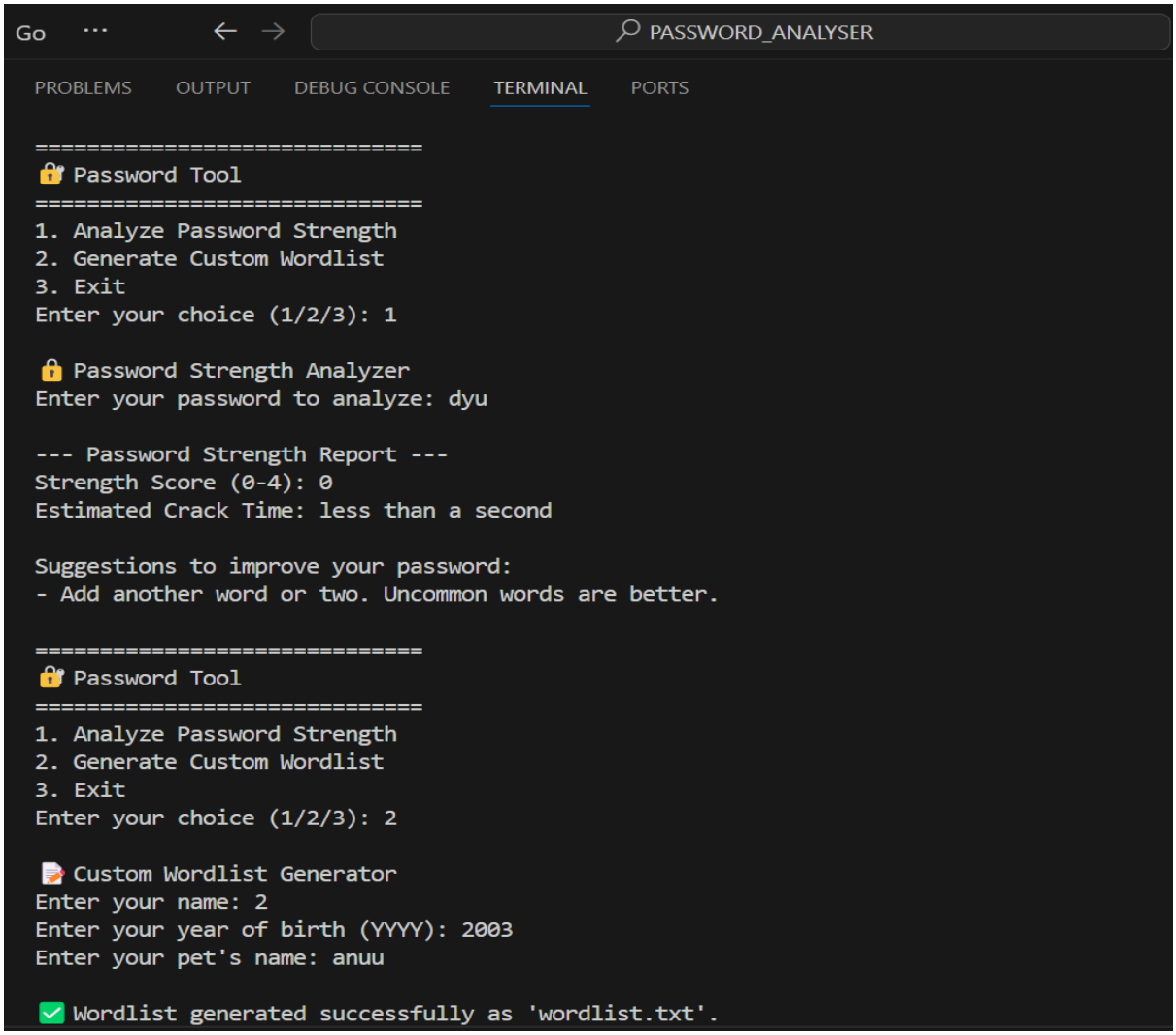
Sample outputs include: -

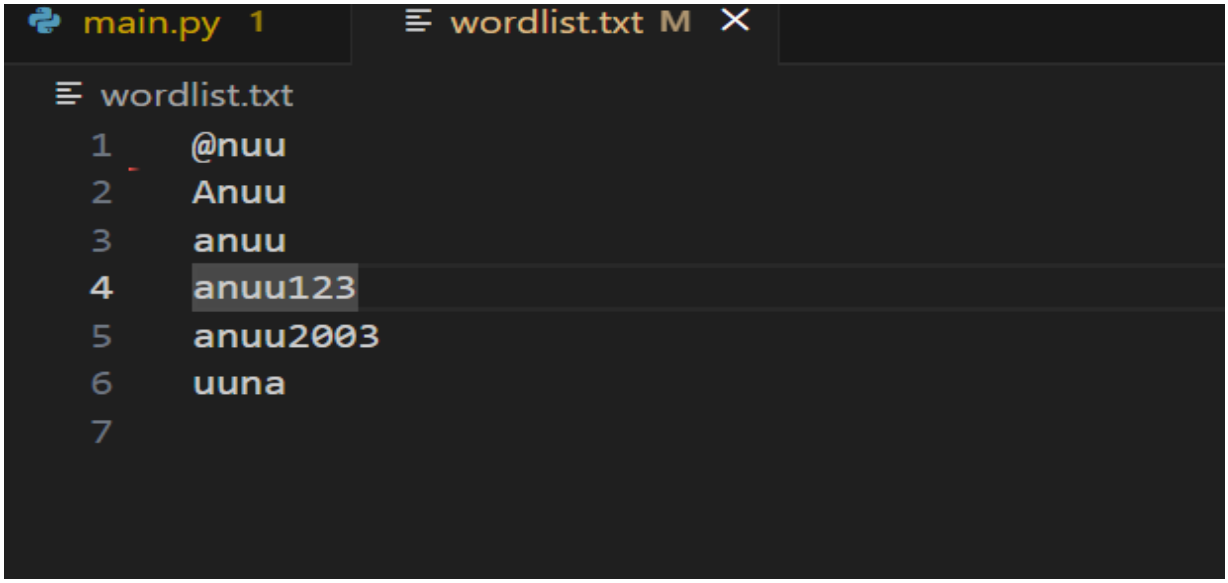
- Password strength score (0-4).
- Estimated crack time.
- Suggestions to improve password strength.
- A generated wordlist saved in wordlist.txt .

RESULTS


Go
PASSWORD_ANALYSER
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
==============================
🔐 Password Tool
==============================
1. Analyze Password Strength
2. Generate Custom Wordlist
3. Exit
Enter your choice (1/2/3): 1
🔒 Password Strength Analyzer
Enter your password to analyze: MyPassword123!
--- Password Strength Report ---
Strength Score (0-4): 3
Estimated Crack Time: less than a second
✅ Your password is strong!
==============================
🔐 Password Tool
==============================
1. Analyze Password Strength
2. Generate Custom Wordlist
3. Exit
Enter your choice (1/2/3): 2
📝 Custom Wordlist Generator
Enter your name: chinnu
Enter your year of birth (YYYY): 2003
Enter your pet's name: kitu
✅ Wordlist generated successfully as 'wordlist.txt'.

RESULTS


Go
PASSWORD_ANALYSER
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
==============================
🔐 Password Tool
==============================
1. Analyze Password Strength
2. Generate Custom Wordlist
3. Exit
Enter your choice (1/2/3): 1
🔒 Password Strength Analyzer
Enter your password to analyze: dyu
--- Password Strength Report ---
Strength Score (0-4): 0
Estimated Crack Time: less than a second
Suggestions to improve your password:
- Add another word or two. Uncommon words are better.
==============================
🔐 Password Tool
==============================
1. Analyze Password Strength
2. Generate Custom Wordlist
3. Exit
Enter your choice (1/2/3): 2
📝 Custom Wordlist Generator
Enter your name: 2
Enter your year of birth (YYYY): 2003
Enter your pet's name: anuu
✅ Wordlist generated successfully as 'wordlist.txt'.


main.py 1
wordlist.txt M
wordlist.txt
1 @nuu
2 Anuu
3 anuu
4 anuu123
5 anuu2003
6 uuna
7

CONCLUSION

The Password Strength Analyzer with Custom Wordlist Generator is a practical tool to promote cybersecurity awareness. It demonstrates the importance of strong passwords and how personal information can make passwords vulnerable. Future improvements could include integrating a GUI and expanding wordlist generation with more complex algorithms.