# Password Strength Analyzer with Custom Wordlist Generator

**Introduction:** In the modern digital landscape, password-based authentication is the first line of defense for protecting user data. Weak passwords can be easily guessed or cracked, leading to data breaches. This project addresses this issue by building a Password Strength Analyzer along with a Custom Wordlist Generator. The analyzer provides feedback on the strength of user-entered passwords, while the generator creates a list of probable passwords based on personalized inputs, which can also be used for ethical password audits.

**Abstract:** This project integrates two essential functionalities aimed at enhancing password security awareness and testing. The Password Strength Analyzer evaluates the robustness of user-entered passwords using entropy-based calculations and advanced password-cracking estimators to provide real-time feedback on their security level. In parallel, the Custom Wordlist Generator creates targeted password lists by accepting user-specific inputs such as names, dates of birth, and other personal keywords—mimicking common strategies used by attackers. Together, these tools not only help users build stronger passwords (defensive approach) but also aid ethical hackers and cybersecurity learners in understanding how weak, guessable passwords can be exploited (offensive approach), making it a valuable dual-purpose utility in cybersecurity education and training.

## Tools :

- Python – Main programming language.
- zxcvbn – A password strength estimation library by Dropbox.
- NLTK – Natural Language Toolkit used for word tokenization and processing.
- Tkinter – Used to create a simple GUI for user interaction.
- argparse – Enables CLI use for advanced users.
- os & subprocess – Handle system-level operations.

## Steps Involved in Building the Project

### 1. Designing the Analyzer Module

- Integrated zxcvbn to compute password strength.
- Extracted detailed feedback like score (0-4), estimated crack time, and common patterns (repeats, dictionary words, etc.).

### 2. Creating the Wordlist Generator

- Collected user inputs such as:
    - Names
    - Important dates

    - Favorite phrases
- Generated permutations, appending digits and symbols.
- Ensured output was written to a .txt file for use with tools like Hydra or John the Ripper.

## 3. Developing GUI with Tkinter

- Built a tabbed interface using ttk.Notebook.
- One tab for analyzing password strength.
- One tab for generating the custom wordlist.
- Included progress bars, alerts, and file save options.

## 4. Adding Logging and Unique ID Tracking

- Each password analysis was timestamped and stored with a unique session ID.
- Helpful for audit trails or training purposes.

## 5. Testing and Validation

- Tested with weak, moderate, and strong passwords.
- Validated output wordlists with sample cracking tools (in ethical settings).

## Conclusion

This project demonstrates the importance of proactive password analysis and user-centric wordlist creation. By helping users understand the weaknesses in their passwords and how attackers can guess based on personal information, the tool promotes stronger security habits. The dual nature—strength evaluation and custom wordlist generation—makes it a valuable educational tool in cybersecurity awareness and training programs.