Project Report - Password Strength Analyzer with Custom Wordlist Generator

---

➢ Introduction

Passwords are often the weakest link in cybersecurity. This project, **Password Strength Analyzer with Custom Wordlist Generator**, evaluates password strength, suggests stronger alternatives for weak passwords, and generates custom wordlists for penetration testing. The tool is simple, educational, and practical for cybersecurity awareness.

➢ Abstract

The Password Strength Analyzer is built using **Python and Flask** with a web-based interface. It integrates the **zxcvbn library** to evaluate password strength and provide actionable feedback. If a password is weak, the tool suggests a randomly generated strong password that includes letters, digits and symbols.

The application also generates custom wordlists from user inputs by applying transformations and saves them as .txt files for download. The interface is designed with **HTML and CSS**, featuring a clean, user-friendly layout.

➢ Objective

- **Analyze password strength** using the zxcvbn library
- **Provide actionable feedback** and suggest strong password
- **Generate custom wordlists** based on user input like names, dates, pets.
- **Export generated wordlists** in .txt format for use with password cracking tools.
- **Build web-based interface** using Flask and HTML/CSS.

➢ Tools Used

- **Languages & Libraries:** Python, zxcvbn, NLTK (optional).
- **Web Framework:** Flask.
- **Frontend:** HTML, CSS for UI styling.
- **IDE:** Visual Studio Code.
- **Version Control:** Git and GitHub.

➢ Methodology

- **Password Analysis**

The password entered by the user is analyzed with the zxcvbn library, which returns a score (0–4) and crack time estimates. If the password is weak, the tool generates and suggests a strong random password

- **Wordlist Generation**

  User-defined inputs (e.g., names, pets) are transformed into variants using:
  - **Leetspeak:** Replacing characters (e.g., a → 4, e → 3).
  - **Year Appending:** Adding years to inputs (e.g., John2000).

  The resulting list is saved to static/web_wordlist.txt for download.

- **Web Interface**

  A simple Flask-based web UI displays:
  - Password score and feedback.
  - Suggested strong password (if weak).
  - Download link for the custom wordlist.

➢ Key Features

- Password scoring and crack time estimation.
- Strong password suggestion when required.
- Custom wordlist generation and downloadable file.
- Simple and responsive design.

➢ Applications

- **Cybersecurity Training:** Educating users on password vulnerabilities.
- **Penetration Testing:** Generating attack-specific wordlists.
- **User Awareness:** Encouraging secure password practices.

➢ Conclusion

This project combines password strength evaluation with custom wordlist generation, making it a valuable tool for cybersecurity learners and professionals. By providing real-time feedback, it promotes the creation of strong passwords and demonstrates practical aspects of ethical hacking.