Project Report: Password Strength Analyzer with Custom Wordlist Generator (Streamlit GUI)

Developer: Vishal Prajapati

Course: B.Tech CSE (AI & ML)

Institute: Krishna Institute of Engineering and Technology, Ghaziabad

Objective

To build a secure, browser-based tool that evaluates password strength using both entropy and rule-based validation. Additionally, it generates a custom password wordlist from user inputs (e.g., name, pet, DOB) for ethical hacking or penetration testing.

Technologies Used

- Python – Core programming logic
- Streamlit – Lightweight web UI framework
- zxcvbn – Entropy-based password analyzer
- regex (re) – Password rule matching
- requests – For API call to HaveIBeenPwned
- hashlib – SHA-1 hashing of passwords
- io – To generate downloadable wordlist

Key Features

1. Password Strength Analysis

- Uses zxcvbn to calculate a base score (0–4)
- Converts score to percentage: (zxcvbn_score + 1) * 10 + (passed_rules / 9) * 50
- Displays result using a progress bar and color-coded feedback (Strong, Moderate, Weak)

2. Rule-Based Validation (R1 to R9)

Rule ID	Description
R1	Minimum 8 characters required
R2	At least one uppercase letter
R3	At least one lowercase letter
R4	At least one digit
R5	At least one special character (@#$%^&*)
R6	Avoid common patterns like '123', 'admin'

R7	Avoid repeated characters like 'aaa'
R8	Avoid sequential characters like '1234', 'abcd'
R9	Do not include your name/email in the password

3. Breach Check

- SHA-1 hash of the password is generated
- Only the first 5 characters sent to HaveIBeenPwned API (privacy-safe)
- Returns how many times the password was found in known data breaches

4. Custom Wordlist Generator

- Accepts comma/space-separated input (e.g., name, pet, hobby, DOB)
- Generates permutations like:
  - name, name123, Name@123, 321eman, N@me2025
- Adds leetspeak, capitalized, reversed, and numeric patterns
- Wordlist is displayed and exported as .txt file
- Useful for password testing with tools like Hydra, JohnTheRipper, etc.

Output Example

Input Password: Dragon#9876
Strength: 91.11%
Breach Status: ✅Not found
Suggestions: Avoid sequential patterns like 1234, abcd

Input Base Words: vishal, simba, 2001

Sample Wordlist:

```
vishal
Vishal
vishal123
simba@123
@simba
2001
123simba
```

How to Run

Install requirements:

```
pip install streamlit zxcvbn requests
```

Run the app:

```
streamlit run password_app.py
```

Benefits

- Real-time password strength analysis
- Checks password leaks using real-world data
- Generates custom wordlists for professional use
- Simple, lightweight web UI (no login needed)
- Privacy-safe (no full password sent externally)

Future Scope

- Add built-in password generator
- Export report as PDF or JSON
- Cloud deployment (Streamlit Cloud / Vercel)
- Multi-language interface
- Auto-email report feature

Conclusion

This tool effectively combines password policy validation, strength analysis, data breach verification, and wordlist generation into one fast and secure application. It's especially useful for awareness, self-testing, and ethical hacking purposes. With a clean UI and real-time feedback, it educates users while helping improve password habits.