**Introduction**

This project is a Python-based tool that analyzes password strength and generates a custom wordlist from user-provided inputs like name, pet name, and dates. It is designed for educational use in cybersecurity to demonstrate how weak or predictable passwords can be exploited.

**Objectives**

* Evaluate password strength based on length and complexity.
* Generate personalized wordlists with variations.
* Save results in .txt format for testing.

## ****Tools & Technologies****

* **Python 3** – Programming language used.
* **argparse** – For command-line argument parsing.
* **datetime** – For generating year variations.
* **re (Regular Expressions)** – For pattern matching in passwords.

**Methodology**

* **Password Analysis**: Checks if a password is at least 8 characters and contains uppercase, lowercase, digits, and symbols.
* **Wordlist Generation**: Uses names, pets, and dates; applies leetspeak substitutions (e.g., a→4, e→3); appends/prepends years (±10 years).
* **Output**: Results are saved to a text file for further use.

## ****Sample Execution****

### Input:

python tool.py --password HelloWorld123! --name Mannat --pet Simba --date 2001 --output mylist.txt

**Results**

* Passwords are categorized as **Weak**, **Medium**, or **Strong**.
* Custom wordlist is generated with hundreds of entries based on user inputs.

## ****Limitations****

* The password strength check is **basic**; it does not calculate entropy like advanced tools (e.g., zxcvbn).
* The wordlist generator is limited to leetspeak and year combinations; no advanced mangling rules are applied.
* Only CLI interface is supported; GUI not implemented.

**Conclusion**

The tool highlights the risks of weak passwords and demonstrates how attackers can use personal details to build wordlists. It emphasizes the importance of strong, unique passwords in securing digital identities.