# **INTERNSHIP FINAL PROJECT REPORT**

NAME: PS JYOSHNA SAI

College Name: SREENIVASA INSTITUTE OF TECHNOLOGY AND MANAGEMENT STUDIES

University: JNTUA University

**Project Title: Password Strength Analyzer With Custom Wordlist Generator**

**1.Introduction**

Passwords are the primary method of securing digital identities. However, weak or predictable passwords can easily be cracked. The aim of this project is to build a tool that not only evaluates the strength of user passwords but also creates a personalized wordlist that attackers might use. This tool is a step toward understanding how attackers think and preparing better defenses.

**2. Abstract**

This project focuses on building a command-line-based tool in python that analyzes the strength of a given password using the zxcvbn library and generates a custom wordlist from user-provided inputs such as names, dates, and pet names. These generated wordlist can mimic real-world password guessing attempts. Additionally, it includes common transformations like leetspeak, appending years, and reversing strings to make the wordlist more realistic. The wordlist is then exported in .txt format to be used with password cracking tools.

**3. Tools Used**

-> Python 3

-> zxcvbn (for password strength analysis)

-> Argparse

-> NLTK

-> tkinter

**4. Steps Involved in Building the Project**

**Step 1:** Password Strength Analysis

Used the zxcvbn Python library to analyze the complexity and guessability of passwords.

**Step 2:** Collecting Custom Inputs

Collected user inputs like name, birthdate, and pet name.

**Step 3:** Wordlist Generation

-Generated variations like:

Original, reversed

Appended years or numbers ( e.x.,123, 2025)

Leetspeak substitutions( a to @, e to 3)

Combinations (name + year, pet + date)

**Step 4:** Exporting Wordlist

* Exported the generated wordlist to a .txt file (custom\_wordlist.txt) which can be used in password cracking tools.

**Step 5:** Optional GUI/CLI

* Developed a user-friendly CLI with argparse. Optionally, a GUI can be added using tkinter.

**5. Conclusion**

This project helps in understanding how weak passwords are formed and guessed. By simulating the thought process of an attacker and generating targeted wordlists, it highlights the importance of using unpredictable, strong passwords. It also demonstrates the real-world application of cybersecurity principles using Python. The tool can be extended further by integrating more advanced features like password hash cracking, database integration, or cloud deployment.

**6. GitHub Repository**

GitHub Link: **https://github.com/Jyoshna1512/Final-Project-Phase**