🔐 Password Strength Analyzer with Custom Wordlist Generator — Final Project Report (Page 1)

1️⃣ Introduction

The Password Strength Analyzer with Custom Wordlist Generator is a comprehensive security tool designed to address modern password vulnerability challenges. It combines educational strength assessment with practical penetration-testing capabilities to serve both individual users and security professionals.

2️⃣ Abstract

This project delivers a robust Python application integrating password strength analysis (zxcvbn) with intelligent, customizable wordlist generation. The GUI (Tkinter) provides real-time feedback while the generation engine produces targeted dictionaries using personal data, leetspeak, and pattern variations.

🧩 Key Areas & Feature Summary

Area	Feature Summary
🔒 Password Analysis	Score (0-4), crack time estimates, pattern detection, suggestions
🧠 Wordlist Generation	Personal data integration, leetspeak, year suffixes, case/special char variations
📤 Export	Hashcat/John-compatible .txt, size management, progress tracking
💻 Compatibility	Windows, Linux, macOS

🛠️ 3. Tools Used

Tool	Purpose
Python 3.8+	Core programming language
Tkinter	GUI development
zxcvbn-python	Password strength estimation
NLTK	Word variations and NLP
argparse	CLI support
re, itertools, datetime	Pattern generation and utilities

🧭 4. Steps Involved in Building the Project

1.	Phase 1: Core Architecture Design — modular MVC structure and configuration system.
2.	Phase 2: Password Analysis Module — integrated zxcvbn for scoring and feedback.

3.	Phase 3: Wordlist Generation Engine — personal info parsing, leetspeak, year appending (1970–2024).
4.	Phase 4: GUI Development — tabbed interface, real-time analysis, export management.
5.	Phase 5: Advanced Features — color-coded strength meter, pattern detection, bulk export.
6.	Phase 6: Testing & Validation — unit tests, performance and compatibility checks.

🔐 Password Strength Analyzer — Implementation, Examples & Deliverables (Page 2)

6️⃣ Example: Password Analysis Function (Python)

```
def analyze_password_strength(password):
    result = zxcvbn(password)
    return {
        'score': result['score'],
        'feedback': result['feedback'],
        'crack_time': result['crack_times_display'],
        'patterns': result['sequence']
    }
```

7️⃣ Advanced Features

- ⚡ Real-time strength meter with visual cues (implemented in GUI logic)
- 🔍 Pattern-based vulnerability detection (keyboard walks, repeated sequences)
- ⚙️ Customizable generation parameters and bulk export
- 🔗 Compatibility with common cracking tools and formats (Hashcat, John)

8️⃣ Conclusion

The project meets objectives by combining defensive education and offensive testing capability. The modular design supports future enhancements. Deliverables include a functional GUI application, CLI support, and exportable wordlists.

📦 Deliverables Achieved

Item	Description
✅ GUI application	Real-time analysis and wordlist generation
✅ CLI tool	Scriptable generation and export
✅ Export Formats	.txt compatible with Hashcat and John the Ripper

✅ Documentation	User guide and testing report

📌 **Project Status:**	Completed Successfully
📅 **Prepared on:**	October 25, 2025
📞 **Contact:**	fatehali_ (for follow-ups)
📝 **Submitted By:**	Fatehali Abbasali Maknojiya
🎓 **Internship Project Duration:**	2 Weeks
📅 **Year:**	2025
🧑‍🏫 **Mentor:**	Elevate Labs