Project Profile

Project Type	Web Application
Project Title	Password Generator
Tech Stack	- HTML - CSS/SCSS - TypeScript/JavaScript
Deployment Platform	Github Pages

Project Introduction

In this exercise, we developed a **prototype password generator** to explore and validate the design choices involved in creating secure, user-friendly passwords. While the implementation is not intended for production use, it serves as a **minimum viable product (MVP)** for testing core functionality, usability, and security considerations. This approach allows us to **analyze trade-offs, evaluate effectiveness, and draw insights** that are relevant to software design and management.

System Overview

Purpose:
The system is a **password generator prototype** designed to produce customizable passwords for demonstration and validation purposes. Its goal is to explore **usability, flexibility, and basic security features** in a simple software context.

Key Components:

1. **Input Module** – Allows the user to select password length, inclusion of letters, numbers, symbols, and case sensitivity.

2. **Character Pools** – Predefined sets of characters (uppercase, lowercase, numbers, symbols) used for password construction.

3. **Password Generation Engine** – Randomly selects characters from chosen pools and assembles them into a password.

4. **Output Module** – Displays the generated password to the user and optionally allows regeneration.

Workflow:

- The user specifies preferences via the input module.
- The generation engine validates inputs and randomly constructs a password according to selected options.
- The output module presents the password and logs generation for validation/testing purposes.

Status & Scope:

- Functioning as an **MVP/prototype**, the system is suitable for testing and validation but is **not production-ready**.
- Focus is on **demonstrating design decisions, usability, and password variability** rather than full-scale security enforcement.

Proposed System

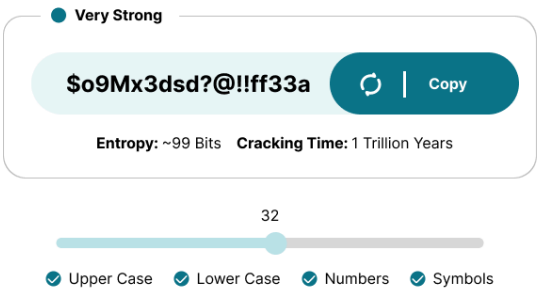
Objectives:

1. **Demonstrate Feasibility of the Password Generator**
    - Show that a modular, customizable password generation system can be implemented as a working prototype.
    - Validate that the core functionality (length, character type selection, randomness) works as intended.
2. **Perform Technical Testing and Validation**
    - Evaluate password strength using **entropy (bits)** and estimated crack time metrics.
    - Assess usability, responsiveness, and correctness of the system under different input scenarios.
3. **Analyze Security Implications**

- Examine how design choices (character sets, length, randomness source) affect password strength.
- Provide qualitative ratings (Very Weak → Very Strong) to support validation results.

4. **Demonstrate Modular and Maintainable Design**
   - Highlight separation of responsibilities across modules (generator, stats, UI).
   - Ensure that the system is extendable for additional features without modifying core logic.
5. **Provide Insights for Software Management Decisions**
   - Use the prototype as a case study for evaluating trade-offs between **usability, complexity, and security**.
   - Showcase how testing and validation inform future improvements or production readiness.

Project Mockup



Testing & Recommandation

Testing Approach:
The password generator prototype underwent **manual testing** to validate functionality, usability, and security metrics. Tests included:

1. **Functionality Testing**
    - Generated passwords of varying lengths and character types.
    - Verified inclusion of selected character sets in all generated passwords.
    - Tested UI features: slider adjustments, checkbox selections, refresh, and copy-to-clipboard.
2. **Security Metrics Testing**
    - Calculated **entropy (bits)** and estimated **crack time** for different password configurations.
    - Confirmed that longer passwords with diverse character sets consistently resulted in higher entropy and stronger passwords.
3. **Usability Testing**
    - Observed interface clarity, responsiveness, and intuitiveness during manual interaction.
    - Confirmed that strength indicators, color codes, and metrics provide clear feedback.

Key Observations:

- Manual testing confirms that the prototype **works reliably** and meets intended functional and usability requirements.
- Password strength scales predictably with **length and character diversity**, supporting validation objectives.
- Current implementation works well as a **demonstration and validation tool**.

Recommendations for Future Improvements:

1. **Responsive Design**

- Optimize the interface for different screen sizes to improve accessibility and user experience.

2. **Maintainability Enhancements**

    - Refactor code for easier updates and integration of new features, while maintaining modular design principles.

    - Consider adding automated tests for regression and performance validation.

3. **Security & Usability Enhancements**

    - Integrate additional security checks such as dictionary word detection.

    - Improve user guidance through tooltips or preset configurations for common password policies.

Conclusion:

The prototype demonstrates **successful feasibility and technical validation**, functioning correctly in manual testing. Future improvements in **responsiveness, maintainability, and enhanced metrics** will strengthen its value as a **demonstration and testing tool** for software management evaluation.