Random Password Generator (RPG)

Description:
The Random Password Generator is a simple Python script that generates secure and random passwords and passphrases. It can be used to create strong and unique passwords and passphrases for various online accounts and enhance your digital security. This project was inspired by the need for stronger and more secure passwords in today's digital world. It is a simple and effective way to enhance your overall online security posture. In summary, strong passwords act as a crucial barrier to protecting your personal and financial information, preventing identity theft, and maintaining your privacy in an increasingly digital and interconnected world.
There are 4 important types of requirements, which are explained below.

Functional Requirements

Aa Name	Assign	Date	Status
Strength Indicator		@January 25, 2024 → January 29, 2024	Not started
Secure Storage Option		@January 30, 2024 → February 3, 2024	Not started
Clipboard Integration		@February 1, 2024 → February 5, 2024	Not started

Aa Name	Assign	Date	Status
Password Policy Compliance		@February 9, 2024 → February 13, 2024	Not started
Batch Generation		@February 17, 2024 → February 21, 2024	Not started
Configuration Options		@January 23, 2024 → January 27, 2024	Not started
Platform Compatibility		@February 2, 2024 → February 6, 2024	Not started
Logging and History		@February 13, 2024 → February 17, 2024	Not started
Integration with Other Tools		@February 15, 2024 → February 19, 2024	Not started
Password Length			Done
Character Set Customization			Done
Randomness Generation			Done
Entropy Source			In progress
Passphrase Support			In progress

FUNCTIONAL REQUIREMENTS

☑ ~~Randomness Generation:~~

- The software should be able to generate truly random and unpredictable passwords or passphrases.

☑ ~~Password Length:~~

- Users should be able to specify the desired length of the generated passwords or passphrases.

☑ ~~Character Set Customization:~~

- Allow users to customize the character set used in the generated passwords, such as including or excluding uppercase letters, lowercase letters, numbers, and special characters.

□ **Passphrase Support:**

- Provide the option to generate passphrases composed of random words or a combination of words and characters.

□ **Strength Indicator:**

- Display a strength indicator or score for generated passwords or passphrases to help users understand the level of security.

□ **Entropy Measurement:**

- Include a feature to measure and display the entropy of generated passwords, indicating the randomness and strength of the password.

□ **Secure Storage Option:**

- Allow users to securely store generated passwords or passphrases, perhaps in an encrypted vault or keychain.

□ **Clipboard Integration:**

- Enable users to easily copy generated passwords or passphrases to the clipboard for convenient pasting into other applications.

□ **Batch Generation:**

- Provide the ability to generate multiple passwords or passphrases in a single operation for bulk use or password changes.

□ **Configuration Options:**

- Include various configuration options, such as the ability to enforce certain criteria (For example, minimum number of uppercase letters) or exclude specific characters.

□ **Platform Compatibility:**

- Ensure compatibility with different operating systems and platforms, making the software accessible to a broad user base.

□ **Password Policy Compliance:**

- Allow users to adhere to specific password policies or guidelines set by organizations, including minimum and maximum length requirements.

□ **Entropy Source:**

- Implement a secure source of entropy for random number generation, ensuring that the generated passwords are truly random.

□ **Logging and History:**

- Optionally, provide a logging or history feature to track the generation of passwords or passphrases for audit purposes.

□ **Integration with Other Tools:**

- Allow integration with other security tools or password management solutions.

Non-Functional Requirements

Name	Assign	Date	Status
Interoperability		@January 24, 2024 → January 28, 2024	Not started
Logging and Auditing		@January 26, 2024 → January 30, 2024	Not started
Scaleability			In progress
Performance			Done
Useability			Done
Security			Done
Reliability			Done
Compatibility			Done
Maintainability			Done

NON-FUNCTIONAL REQUIREMENTS

☑ ~~**Performance:**~~

- *Response Time:* The software should generate passwords or passphrases quickly, providing a responsive user experience.
- *Scalability:* The tool should perform well, even with a large number of users or when generating multiple passwords concurrently.

☑ ~~**Security:**~~

- *Randomness:* The generated passwords or passphrases should exhibit a high degree of randomness to resist predictability.
- *Entropy:* The software should use a reliable source of entropy for random number generation to enhance the security of generated passwords.
- *Secure Storage:* If the tool includes a password storage feature, it should store passwords securely, possibly using encryption and access controls.

☑ ~~**Usability**:~~

- *User Interface:* The user interface should be intuitive, providing clear instructions and options for users to customize their password or passphrase preferences.
- *Accessibility:* The software should be designed to be accessible to users with disabilities, following accessibility standards.

☑ ~~**Reliability**:~~

- *Stability:* The tool should be stable and reliable, minimizing the likelihood of crashes or errors during operation.
- *Error Handling:* The software should gracefully handle errors, providing meaningful error messages to users.

☑ ~~**Compatibility**:~~

- P*latform Compatibility:* The software should be compatible with various operating systems and environments.
- *Browser Compatibility:* If the tool has a web-based interface, it should be compatible with different web browsers.

☑ ~~**Maintainability**:~~

- *Code Maintainability:* The software code should be well-organized and documented, facilitating future maintenance and enhancements.
- *Configurability:* The tool should allow for easy configuration and updates, especially when changing parameters like password complexity rules.

□ **Scalability**:

- *Number of Users:* The software should be scalable to accommodate an increasing number of users.
- *Volume of Requests:* It should handle a high volume of password or passphrase generation requests efficiently.

□ **Interoperability**:

- *Integration:* If the tool is part of a larger system, it should integrate smoothly with other components and tools.

□ **Logging and Auditing**:

- *Logging:* The tool should have effective logging mechanisms to record important events for troubleshooting and auditing purposes.

Transition Requirements

Aa Name	Assign	Date	Status
Installation Instructions		@January 23, 2024 → January 27, 2024	Not started
Release Notes		@January 30, 2024 → February 3, 2024	Not started
License Management		@January 30, 2024 → February 3, 2024	Not started
Security Considerations		@January 22, 2024 → January 26, 2024	Not started
Monitoring and Evaluation		@February 7, 2024 → February 11, 2024	Not started
Deployment Planning		@December 29, 2023 → January 2, 2024	Done
Configuration Management		@January 6, 2024 → January 10, 2024	Done
Rollback Plan		@January 15, 2024 → January 19, 2024	In progress

TRANSITION REQUIREMENTS

☑ ~~Deployment Planning:~~

- Define a deployment plan that outlines the steps for installing and configuring the password generator software.
- Specify any dependencies or prerequisites for successful deployment.

□ **Installation Instructions:**

- Provide clear and concise installation instructions for administrators or end-users.
- Include information about system requirements, supported platforms, and any third-party dependencies.

☑ ~~**Configuration Management:**~~

- Establish a configuration management process to track and manage changes to the software configuration.
- Implement version control mechanisms to handle different releases of the software.

□ **Rollback Plan:**

- Develop a rollback plan in case issues arise during deployment.
- Specify the steps to revert to the previous version or state if necessary.

□ **Release Notes:**

- Create comprehensive release notes that document changes, new features, bug fixes, and any known issues.
- Share release notes with users and administrators.

□ **License Management:**

- Ensure that license information is managed appropriately, and users are aware of licensing terms.
- Guide how to update or renew licenses, if applicable.

□ **Security Considerations:**

- Conduct a security review to ensure that the transition does not introduce vulnerabilities.
- Update security documentation and guidelines.

□ **Monitoring and Evaluation:**

- Implement monitoring tools to track the performance and usage of the password generator in the production environment.
- Schedule evaluations to assess the effectiveness of the transition process and gather feedback for improvement.

Interface Requirements

Name	Assign	Date	Status
Web Interface		@January 29, 2024 → February 2, 2024	Not started
Multi-Language Support		@January 24, 2024 → January 28, 2024	Not started
Help and Documentation		@February 13, 2024 → February 17, 2024	Not started
Integration with Password Managers		@January 21, 2024 → January 25, 2024	Not started
Integration with Browser Extensions		@January 29, 2024 → February 2, 2024	Not started
API Support		@January 31, 2024 → February 4, 2024	Not started
Cross-Platform Compatibility		@February 3, 2024 → February 7, 2024	Not started
User Interface			Done
Command-Line Interface			Done
Error Handling			In progress
Customization Options			In progress

INTERFACE REQUIREMENTS

☑ ~~User Interface:~~

- The software should have an intuitive and user-friendly graphical user interface (GUI) for users to interact with.
- The UI should provide options for users to customize password or passphrase parameters, such as length and character sets.

☑ ~~Command-Line Interface:~~

- If applicable, the software should support a command-line interface for users who prefer a text-based interaction or for integration with scripts and automation.

□ **Web Interface:**

- For web-based versions, provide a clean and responsive web interface accessible through popular web browsers.
- Ensure compatibility with major browsers such as Chrome, Firefox, Safari, and Edge.

□ **Multi-Language Support:**

- Support multiple languages in the user interface to accommodate users from different regions.

□ **Customization Options:**

- Provide users with the ability to customize the appearance of the interface, such as themes or color schemes.

□ **Error Handling:**

- Implement clear and informative error messages to guide users in case of input errors or other issues.
- Include error prevention mechanisms to minimize user mistakes.

□ **Help and Documentation:**

- Include built-in help features within the interface to guide users on how to use the software effectively.
- Provide online documentation or tooltips for quick reference.

□ **Integration with Password Managers:**

- Allow users to easily copy generated passwords or passphrases to a clipboard or integrate with popular password management tools.

□ **Integration with Browser Extensions:**

- If applicable, provide browser extensions or plugins to integrate the password generator with web browsers.

□ **API Support:**

- If the software is intended to be integrated into other applications or systems, provide an Application Programming Interface (API) for programmatic access.

□ **Cross-Platform Compatibility:**

- Design the interface to be compatible with various operating systems (Windows, macOS, Linux) without sacrificing usability.