Manual (revised version)

Feature 1 - Password Length

User Tutorial
When prompted, input the number of characters you would like the password to be lengthwise. The program will then generate a password exactly matching the length specified. Users should be aware that longer passwords generally provide better security.

How it Works
In the code, the password length is determined by the user's input in program.py. The length variable is set by asking the user to input the desired password length, which is converted to an integer. This length is then passed as an argument to the generate_password function in password_generator.py. The function uses this length to ensure the generated password matches the specified number of characters.

Feature 2 - Strength of Randomness

User Tutorial
When prompted, type either ‘low’, ‘medium’, or ‘high’ for how strong you want the password’s randomness to be. Users can select the desired strength of their password, based on their security needs. A high-strength password is recommended for better security, especially for sensitive accounts.

How it Works
Password strength is set based on user input for the desired level ('low', 'medium', or 'high') in password_utils.py using the ask_strength function. This input is passed to the generate_password function in password_generator.py. The strength level influences the password generation logic.

Feature 3 - Pronounceability

User Tutorial
Users opting for pronounceable passwords should expect passwords that are easier to remember and say aloud, unlike typical random-character passwords. This feature is useful for users who prefer passwords that are not just random strings of characters but have some semblance of readability.

How it Works
The pronounceability feature is controlled by the user's input in program.py, asking if they want a pronounceable password. If yes, the generate_password function in password_generator.py calls generate_pronounceable_password from markov_chain.py. This function uses a Markov

chain, built from a predefined set of English words, to generate a password that resembles real words, thus making it more pronounceable.

Feature 4 - Password Entropy

User Tutorial
After generating a password, users will receive information about the password's entropy. High entropy indicates a strong password. Users should use this information to gauge the relative security of their password, with higher entropy typically signifying better security.

How it Works
The password entropy calculation is performed in password_generator.py by the calculate_entropy function. After a password is generated, this function is called to calculate its entropy. The entropy calculation uses the logarithm of the number of possible combinations of characters in the password (calculated as the character space to the power of the password length).

Feature 5 - Custom Character Set

User Tutorial
When the prompt comes up that allows you to specify a custom set of characters, type in ‘yes’. If this option is chosen, the program prompts the user to input their desired characters, which are then used in the password generation process. This feature provides flexibility in creating a password that adheres to specific rules or preferences.

How it Works
If the user opts to use custom characters (determined in program.py), they are prompted to enter them. These characters are then passed to the generate_password function, which ensures that only these characters are used in the password generation process. The generate_password function either passes these characters to generate_pronounceable_password in markov_chain.py for a pronounceable password, or uses them directly in generate_random_password() for a random password.