Student Information

- o beesleys1@mymail.nku.edu
- 0 (859) 412-7274
- o GitHub Repository: https://github.com/ScramblesDev/ASE420-Individual-Project

Planning the Individual Project

Summary of the Project

I want to create a program in terminal that generates secure passwords for users.

Goals

My goal is to create a password generator that a user can run right from their command-line. This will give them numerous options (such as character length, character type, etc). I would like to have first-hand experience with creating something that helps people in terms of security. I will learn software design fundamentals and apply them to my team project as well.

Requirements (User Stories)

User Story 1: Setting Password Length

 As a user who values security (Role), I want to be able to specify the length of the generated password, so that I can have control over how long or short my passwords are. The app should allow me to enter a numeric value for the desired password length. Passwords generated should adhere to the specified length.

User Story 2: Low/Medium/High Complexity Password

As a user who also values security, I want the option to generate different levels
of complexity for my password, so that I can use it for different levels of critical
accounts where security's importance is variable. The app should provide
multiple options for password complexity.

User Story 3: Pronounceable Passwords

 As a user who prefers memorable passwords (Role), I want the option to generate pronounceable passwords (Action), composed of words or phrases, for accounts where I value ease of use (Benefit). The app should provide a "Pronounceable Password" option for password generation. I should be able to specify the number of words or phrases to include in the password.

User Story 4: Custom Character Option

 As a user who requires specific characters in my passwords, I want the option to customize the character set used in password generation, so that I can meet the unique requirements of certain websites or services. The app should provide a "Custom Characters" option for password generation. I should be able to enter or select specific characters, symbols, or character classes to include in the password.

• User Story 5: Entropy Report

As a security-conscious user, I want the app to provide an entropy report for the generated password, so that I can assess the strength and randomness of the password. After generating a password, the app should display an entropy report in the form of a number Users should be able to understand the password's complexity based on the entropy value.

Features

Password Length Control

 Allow users to specify the length of the generated password, ensuring it meets the requirements of different services or systems.

Character Set Customization

 Provide options to include or exclude different character sets, such as uppercase letters, lowercase letters, numbers, and special symbols.

Password Strength Indicator

 Display a password strength indicator that evaluates the generated password and provides feedback on its strength (e.g., weak, medium, strong).

Pronounceable Passwords

 Generate passwords that are easy to pronounce or remember for users who prefer such passwords.

Randomness Control

 Offer settings to control the randomness of the password, allowing users to adjust the level of complexity

Deliverables

Design: Link

Code: Link

Test: Link

Documents: Link

Project Test Plan

- Character Set Testing: Verify that the application correctly defines and uses character sets for different types of characters (uppercase, lowercase, numbers, symbols).
- **Password Length Testing**: Test the application to ensure that it generates passwords of the specified length.
- Randomness Testing: Check that the generated passwords exhibit sufficient randomness and don't show patterns or biases.

Milestones and Deadline

Milestone 1

- Expectation Date: Nov 5
- Goal: Set up the project environment and create the basic structure of the application, including the main program file and necessary dependencies.

Milestone 2

- Expectation Date: Nov 10
- Goal: Implement password length control, allowing users to specify the length of generated passwords.

Milestone 3

- Expectation Date: November 16
- Goal: Add character set customization, enabling users to select which character sets (uppercase letters, lowercase letters, numbers, symbols) to include in passwords.

Milestone 4

- Expectation Date: November 20
- Goal: Develop the password strength indicator, which evaluates generated passwords and provides feedback on their strength (e.g., weak, medium, strong).

Milestone 5

- Expectation Date: November 26
- Goal: Implement the generation of pronounceable passwords, providing an option for users who prefer easily pronounceable passwords.

Milestone 6

Expectation Date: November 29

• Goal: Perform thorough testing, bug fixing, and documentation to prepare the application for the main deadline on December 1.

Risk Analysis

Monday: Available 2pm - 4:40pm, 9pm - 11:59pm

Tuesday: Available 3pm - 11:59pm

Wednesday: Available 2pm - 4:40pm, 9pm - 11:59pm

Thursday: Available 9pm - 4am

Friday: Unavailable

Saturday: Unavailable

Sunday: Available 7:30pm - 11:59pm

Note: Let me know if you have any questions! I am flexible and can make time.

Project Progress

Week 1

Summary

Set up the team (and individual) project pages for documentation, and discuss features with team members.

Milestones and Risks

Completed Milestone 1:

Date: November 5th

No risks.

Week 2

Summary

• Have set up the github page, and have begun implementing a basic prototype for the application.

• Have completed a basic function for setting up the password length.

Milestones and Risks

Milestone 2: Completed

No risks.

Week 3

Summary

Have revised the password length generation, and completed basic functionality of all other functions.

- Implemented password length
- Implemented character set customization
- Implemented strength of randomness

Milestones and Risks

Milestone 3: Completed

Risks: Was sick almost the entire week, though that gave me more time to work on my OWN project.

Week 4

<u>Summary</u>

Have revised the code, and added 2 more features:

- Implemented a basic entropy indicator
- Implemented a basic pronounceability function for generating passwords

Milestones and Risks

Milestone 4: Completed

Risks: I'm still sick right now, I ended up getting COVID. I'm going to continue on this still.

Week 5

<u>Summary</u>

Have separated the code into multiple files for modularity. This should make everything EXTREMELY easier. Will begin unit testing next week.

Milestones and Risks

Milestone 5: Completed

Risks: Stressful classes, ADHD-related interferences.

Week 6

<u>Summary</u>

Have finished all features, testing, and documentation.

- Uploaded EVERYTHING to GitHub, including documentation, prototypes, and revisions.
- Completed unit testing with PyTest
- Have made small revisions of the code, and made password_utils.py (to make room for main.py)

Milestones and Risks

Milestone 6: Completed

Risks: More ADHD-related interferences.

Code: Link