Project Charter
Password Generator
Mira L. and Arnav M.

PROJECT STATEMENT

The current issue is the growing vulnerability of online accounts due to weak passwords, which leads to an increase in security breaches and illegal access and calls for the creation of a strong password generator application to strengthen digital security measures and secure user information. The opportunity lies in developing a password generator program that generates strong, unique passwords tailored to individual user preferences, promoting better online security practices, and reducing the risk of data breaches in order to address the inconvenience and security risks associated with passwords that are simple to guess or reuse.

CONCISE PROJECT OVERVIEW

The complexity and size of the project will determine how long it takes to code a password generator. For a simple password generator, 4-6 weeks would be a fair estimate. There is no funding required for the project because we are using our personal computers. At various phases, the project may need the consent of numerous stakeholders. Project commencement, design and architecture, user interface and experience, security measures, and ultimate deployment may all require significant approvals. Project managers, developers, UI/UX designers, security experts, quality assurance teams, and end users who will use the generated passwords are often the major stakeholders for a password generator project. When creating a password generator, one may make assumptions about the complexity or length of the password. Compatibility demands for various platforms, operating systems, and browsers are frequent restrictions while developing password generators. The potential for creating weak passwords is one of the main concerns connected with constructing a password generator.

SCOPE STATEMENT

Creating a computer program that produces strong, one-of-a-kind passwords is part of the scope of coding a password generator. The software must be able to create passwords with a specified complexity level and length. The accomplishment of particular milestones, such as the effective implementation of the password generation algorithm, user interface design, integration of security measures, and testing/validation of the produced passwords, can be used to measure the deliverables for developing a password generator. Given the availability of knowledgeable developers and access to pertinent libraries or frameworks for password generation, coding a password generator is feasible. The password generator project could be finished in three to four weeks, allowing two weeks for testing and bug fixes prior to the official release. In terms of what the scope does not include, it's important to clarify that the password generator project does not encompass the development of an entire authentication system or user management system.

STAKEHOLDERS

The project manager is in charge of supervising every aspect of the code-generation project, leading the team, controlling budgets and resources, and assuring project success. In order to handle any issues or obstacles and guarantee alignment with project objectives, they necessitate regular communication with all stakeholders. Developers are in charge of creating and implementing the password generator's code in accordance with the project's specifications and requirements. For the password generation code, they want precise specifications and prerequisites. To make sure the generator is reliable and secure, security professionals offer advice on how to apply secure coding principles, encryption techniques, and best practices for password generating. To comprehend the security requirements and offer direction on secure coding techniques, they want open communication.

TIMELINE

Project Initiation and Planning (1 day):

- Define project goals, objectives, and scope.
- Identify key stakeholders and their roles.
- Create a project timeline and milestones.
- Allocate necessary resources and set the budget.
- Develop the project plan and obtain necessary approvals.

Requirements Gathering and Analysis (2 days):

- Identify the specific requirements and functionalities of the password generator.
- Analyze user needs and security requirements.
- Document the detailed functional and technical specifications.

Design and Architecture (2 days):

- Design the user experience (UX) of the password generator.
- Develop the architecture and system design for the password generator.
- Define the algorithm for password generation and security measures.

Development (2 days):

- Implement the code for the password generator according to the design and specifications.
- Build the necessary modules and functions for password generation.
- Integrate security measures and encryption techniques.

Testing and Quality Assurance (1 day):

- Conduct functional testing to ensure the password generator functions as intended.
- Perform security testing to identify and address any vulnerabilities.

Documentation (2 days):

- Prepare detailed documentation

BUDGET ESTIMATE(S)

There are no costs. There is one week until the due date and we will be using our own laptops as resources in order to get the project done.

RISKS AND CONTINGENCY PLANS

There is a possibility that there could be a weak password generation algorithm. We can mitigate this by using a proven and secure password generation algorithm, such as a cryptographically secure pseudo-random number generator. There could also be inadequate security measures We can mitigate this by implementing strong encryption techniques for storing and transmitting generated passwords. There could also be insufficient testing. We can mitigate this by implementing a comprehensive testing strategy, including unit testing, integration testing, and user acceptance testing.