Comp-1004: Computing practice 2023/2024.

Password strength checker and password generation application (Theme: Cyber Security)

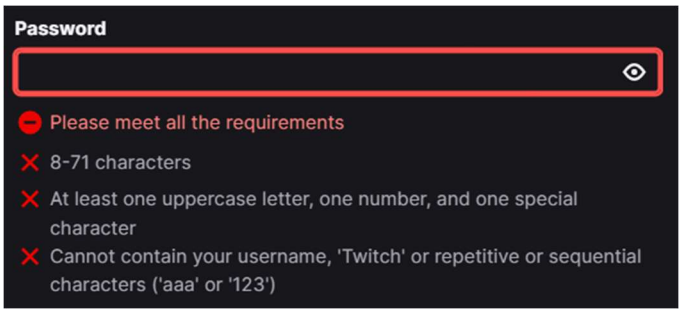
Introduction

With how important the internet has become in daily life and as the number of things that can be done over the internet expands, proper password security has become more important than ever due to cyber crime becoming more common in recent years, mainly focusing on the older generations due to their lack of familiarity with computers and overall cybersecurity knowledge, due to this the passwords used tend to be simple, using peoples names and birthdays, all of which can be found online which makes it easier for people to guess passwords.

Due to this its encouraged to not use names and birthdays in passwords to make it harder for people to get your passwords by using public information that can be gathered through things like Facebook. There are a standard set of rules that websites enforce for passwords, though it can differ, but it tends to all be similar, but some websites can be stricter than others, shown bellow are some examples of some website's password requirements.

1. **Make your Twitch password as unique as you.** Don't reuse the same password across multiple sites. If you have a favorite password that you created for another site, please don't use it here!
2. **Make your password longer and more complex.** If you use a password manager (such as 1Password or LastPass), you're in luck, it can do this for you. If you aren't using a password manager, here's what you need to know. Your passwords should be at least 8-10 characters long, but the longer the better. Your password can be more than one word! It is easier to remember "easier to remember" than Password1, or 987654321 and it is a better password.
3. **Your password can be too easy (it doesn't have enough characters), too weak (not complex enough), just fair (we recommend that you make it longer), or strong.** Choose a combination of letters, numbers, and special characters to make your password stronger. Do not reuse a previous password.
4. **Be creative! Don't use single dictionary words.** Avoid simple words, phrases, and patterns that are easy to guess. Passwords with obvious words and phrases like "password", "twitch", or "1234" put your account at risk.

Source : https://help.twitch.tv/s/article/creating-a-strong-password?language=en_US#:~:text=If%20you%20aren%27t%20using,it%20is%20a%20better%20password.



Source : https://www.twitch.tv/

This first image shows the password requirements and suggestions for Twitch which is a very famous streaming websites, with the 2nd image showing what it enforces when signing up for the site, these show a strong set of base requirements but it can be difficult for some people to make passwords that would fit these requirements, to help with this I will develop a single page application that can help people make passwords by making a password generator and also making a password strength checker so people can check the strength of their passwords based on the criteria set. In this report, I will show the stages of the software development lifecycle that the project went through, along with showing the user requirements, the user stories and the system architecture. It will also contain the sprint plans for the project that show what needed to be done at the various stages of implementation, during this I will evaluate the parts of the project to see what may not be relevant to the project, I will also include diagrams to help people understand the various parts of the project.

Software development lifecycle (SDLC)

The software development lifecycle is important for projects as it helps to keep the project organized throughout its various stages, it tends to be broken down into 5 main stages, these: Requirements, Design, Implementation, Testing, and Maintenance, there are various types of SDLC that follow the 5 main stages but add to them in various ways or alter them so that they better fit the type of projects that they are used for.

Planning

This step focuses on finding out the requirements for the project including the function, nonfunctional, and the usability requirements. With the functional requirements covering anything the project requires to function as needed for the project, the non-functional requirements covering how the project should do it, the usability requirements of the project describe how easy the project is to use overall. There can also be a set of external requirements, this can include things such as social, ethical, and legal considerations for the project.

Design

After the requirements for the project have been documented and decided on, you then move onto the design phase of the project focusing on how the project will look and how it will function, you need to design the basic architecture of the project, during the design phase it is good to create diagrams that help show the project functions such as UML diagrams, state and sequence diagram to help depict things further.

Implementation

During the implementation phase you need to start putting the things you designed into code for the project, this tends to be one of the most important stages of the project as you are writing the code that the project needs to run and if the code is wrong, the project wont work as intended which can extend the projects runtime even further than what may be allowed for the project which can lead to corners needing to be cut for the project to be completed by the deadline. During this project, I will be making use of HTML, CSS, and JavaScript to make it function the way I intend.

Testing

Testing happens at various points throughout the project to ensure that the features that have been developed are working as intended and do not cause issues with the rest of the project that has been developed so far, testing needs to be done regularly to ensure that there are no software breaking errors that will require the developers to go back multiple times to try and fix it and in the process possibly break the code that has been more recently developed or if very severe needing the developers to go back to previous stages to rework the code from the ground up.

Maintenance

The maintenance phases tends to be the end of the software development lifecycle and focuses on post launch maintenance of the project to ensure that it gets any updates needed if the code breaks or if the clients need something added, the code could break for any number of reasons, some examples of this being operating system updated, this stages is about ensuring that the software works well post launch.

Type of SDLC

There are various types of SDLC often referred to as methodologies, some examples of them are the Waterfall, PRINCE2, Agile, and Spiral. These all vary in how they work and what types of project they work for and how they are executed, some are more flexible with their stages

while some would the project to start again if you need to return to an earlier stage, For this project, we have been assigned the Agile methodology to use.

Agile

For this project, I will make use of the Agile methodology which makes use of “Sprints” which are short development cycles which are useful for allowing the team to focus on specific parts of the project and which tend to span roughly 2 weeks with meetings at the beginning and end of each day and at the end of the sprint to see what progress was made and to see what needs to be done after this, during these meetings, they would discuss progress made and any pain points in the project that would need to be addressed or what may needed to be added for the project to function properly. Agile is good for projects where early prototypes are important such as web development which is what the project is, so it works out well. Shown below is a diagram that shows the various stages for the sprints that take place.

User stories and backlogs are frequently used in this type of SDLC, with user stories representing what features the client or the developer may want to the project to contain in a near first person perspective while not all of them have to be fulfilled it gives an idea of what the client wants. The backlog is a document that contains a list of all features that are yet to be implemented, which features are under development and what features have been completed.

Source : https://asana.com/resources/agile-methodology

Design Documentation

Project Description

With the ever-increasing importance that the internet and how much is dealt with online, cyber security has become more important than ever before, a major part of this involving passwords and the importance of having a strong password on all of your accounts. Passwords are one of the most important ways of keeping our data and accounts safe, to

ensure that our passwords for accounts are as safe as possible, most websites have a set of requirements that the passwords must meet in order to be accepted with it being recommended that you shouldn't use words in the dictionary, names, or birthdays to make a strong password as it reduces the efficiency of brute force attacks which just try random combinations to crack passwords. Another example of a requirement is ensuring the password is a good length, this is to increase the number of combinations a computer would need to try to get the password.

As I stated in my introduction, the project will be a single page application focusing on password generation and password strength checking by making use of JavaScript to produce a random string of characters that would fit the requirements set by websites for passwords to help people that have issues when it comes to making passwords for themselves, while also allowing them to check the strength of passwords they are already using by making use of the password strength checker that the website will have, this will also be made possible by using JavaScript to run tests against the password that have been inputted to the site, to make this site easy for users to make use of, I will implement a user friendly UI that and include explanations of each tool on the webpage so that users can understand what each things does.

To make this possible, I will make use of JavaScript to do various things to allow the site to function such as using it to check input against a set of requirements that can be set, and using it to generate passwords according to a set of requirements that can be changed as needed.

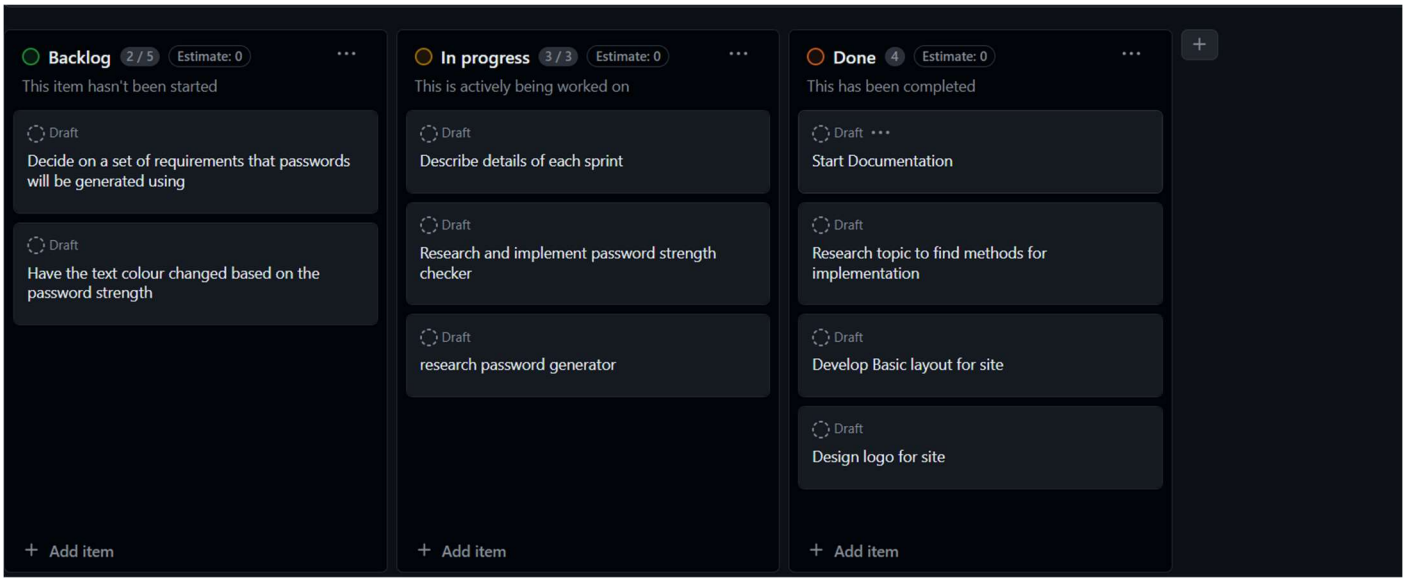
For social requirements, I need to ensure that the site is easy accessible for people to use and not needlessly overcomplication process, and ensuring that the site can easily be used by people that struggle with vision issues, this can be done by not making the text on the screen too small and ensuring that the colours used by the site aren't too strong where it could cause users to have trouble reading the site. An ethical issue is to ensure that passwords that are generated/checked aren't stored as that would highly unethical and quite possibly illegal to do so.

User Stories

- As a user I want the ability to generate passwords that meet the requirements of most modern websites
- As a user I want to be able to easily navigate the page and find the content that I want easily
- As a user I want to be able to see a rough estimation of the strength of my password
- As a user I want to be able to easily tell what the strength of my password by making it easy to be able to see
- As a user, I want to know what each part of the site does.
- As a user I want to be able to what the requirements that my password is being tested against are.

Project backlog

Below shows screen shots of the project backlog with the tasks that were being worked on, that were in progress and that were completed, this shows the various tasks were decided on at that moment in time.



Webpage Design

Basic outline

Shown below is a very rough outline of what the site will look like while possibly using other colour schemes due to development changes.

Position of various site aspects

This is the basic layout the site will use for the main sections that the site will have.

SPRINTS

During the course of the project, there will be many sprints in order to cover the various aspects of the site that need to be developed that will last about 2 weeks and will review the content of the sprint at the end of each sprint, these sprints can be planned out quite early into development and if it is decided what is to be done in each sprint, it can lead to a stable and quick design and planning stage which will make the implementation stages of the project much easier. It also splits the work down into smaller manageable parts which allows the developer to focus on certain aspects of the project at each time making it easier to focus on parts and when each is done they can move onto the next part of the project, for this project we were expected to do 2 sprints before Christmas, while the overall number of sprints will vary between the site and how long it take to develop.

Sprint 1

The first sprint done was focused on the planning stage to ensure that I had all the details to ensure that I could move onto the next stages easier, during this stage I decided what I wanted to do for the project and layout for the design stage, then adding user stories to decide on what features I wanted to add to the site. I made a quick wireframe for the site to show the main sections that would be used. During the stand up meeting, I discussed what my website would be and my thoughts behind it and the

way to layout my designs, I was also told to focus on user stories first, while not all may be achievable its best to add them anyway and it can be changed as needed.

Sprint 2

The second sprint before Christmas included the very start of development, such as creating the basic layout for the site and research into ways to implement the features that I want the site to have and choosing a layout to use for the site as a whole.

At the start of the sprint, I started development by working on the CSS of the page and the outlines I would use and working on the code for the website in html to get the basic elements needed in. things can be changed as needed later.

Sprint 3

After Christmas, I began more development on the site, fixing minor issues that have been found, and ensuring that the site looks how I want it to, and starting research into how to use java script to achieve the goals of the project and deciding on the basics of how I wanted the java script aspects to work, I would also like to finalize the looks of my site so that I can focus on other tasks in the project.

Sprint 4

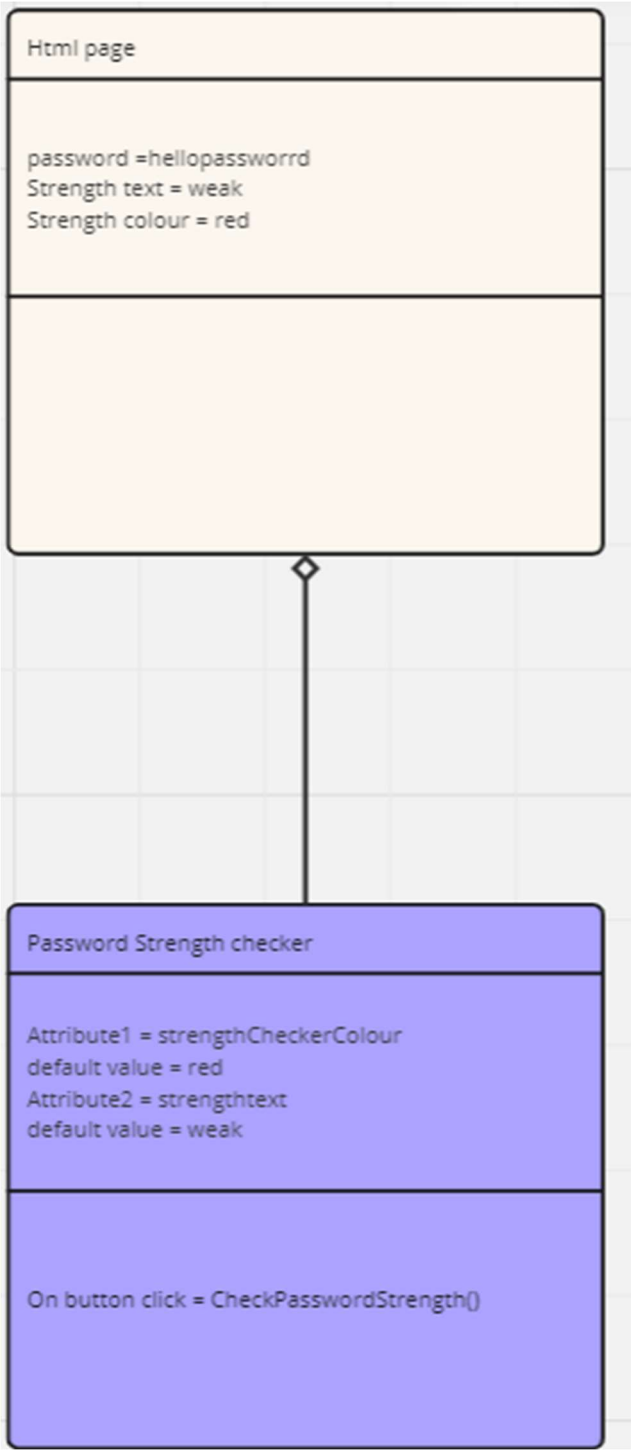
This sprint was focused on working on the JavaScript that would make the password strength checker work and ensure that the features related to it that were mentioned in the user stories would be implemented such as the text changing colour and the text changing depending on which test succeeds.

Sprint 5

Sprint 5 will be one of the final sprints as this sprint focuses on implementing the password generator and the JavaScript necessary for it to run as long with adding html elements to display the password in a way that is easy to read and easy to copy, along with adding other small tasks needed to get the project to a state I'm happy for it to finish in, during the course of this sprint I also had to scrap and redo a lot of work due to the code breaking

UML Diagrams

Below is a UML diagram I made for an aspect of the project.

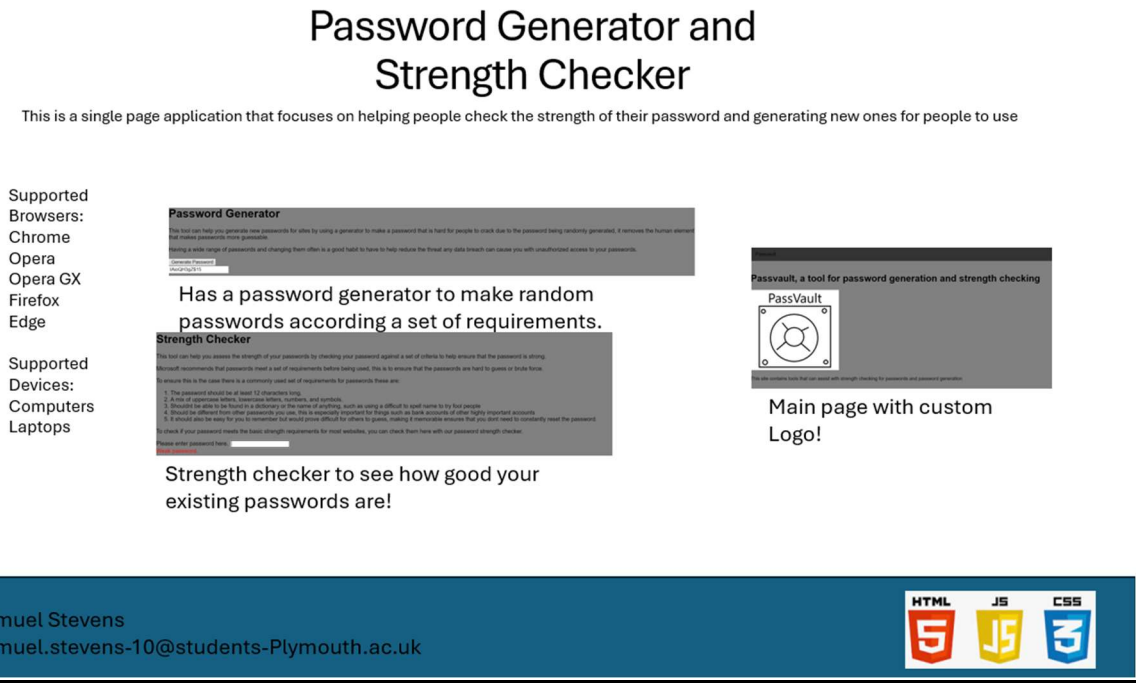


Account of issues and challenges faced.

During the course of the project I face many challenges when it comes to developing the site to work according to the user stories and what I want the site to be, one of the biggest challenges was finding resources online to help teach the skills needed to do the things I wanted and without using server side scripting which is what a lot of JavaScript relies on for various aspects to make things work, this made the work much harder to do as I was spending much more time researching the implementation methods and testing them than was needed to make the various aspects of the project to work, making things

writing and getting information from files much more difficult due to the lack of server side scripting. there was also the issue of needing to change important things right at the end of the project due to it causing things to break which would make it impossible to work on, so a lot of work had to be scrapped so that I can make the rest of the site work as needed, The scenario was also vague on whether use of frameworks were allowed so we had to work around that which caused the development time to be much longer than it could have been by using them.

Poster screenshot



Link to Github Repo: https://github.com/Samuel-Stevens/COMP-1004Project