[Assessment Criteria](https://dle.plymouth.ac.uk/pluginfile.php/3242871/mod_resource/content/0/COMP1004%20-%20Assessment.pdf)

[Scenarios](https://dle.plymouth.ac.uk/pluginfile.php/3242873/mod_resource/content/0/COMP1004%20Project%20Scenarios.pdf)

Independently generated concept: Notepad style application  
Webapp with the ability to write, read, save and load text. This is **insufficient** to get full marks but is passable. Additions will improve value, current addition ideas:

* Formatting (bold, italics, underline, etc) – this can be toggled on or off, and likely uses tags (customisable or generic like <b> </b>?)
* Formatter for code

**~~Alternatively, provided games development scenario:~~**

~~The focus of this project is to explore leisure time and consider what might enrich this. Additionally, the project should also record time spent playing the game and output this into a flat file.~~

~~Produce a hand coded hypertext fiction game using HTML, CSS and JavaScript enhancements. Your game must be playable from start to finish. You will need some custom art for the game pages. There should be a clear narrative structure and plot. The game structure should be coded in a clear, logical manner using JavaScript.~~

~~Your application should output to a flat file the following JSON data format. It should have a player username, minutes the player has spent playing, their score and the level.~~

**Alternatively, provided Cyber Security scenario:**

The focus of this project is to provide a password account management system for a user accessed via a browser.

Produce a hand coded application whereby an individual can manage their own passwords and accounts for different applications. As a minimum your application should provide the following functionality:

* Use local storage to store personal account data such as username and password.
* Read a JSON file using the format above to identify a matching username in your system.
* Indicate password strength to the end user.

You will need to determine what additional features this application should have and the best way to visualise this information for the user.

Encrypt personal information (cannot hash as it needs to be reused) using a high end encryption method.  
Search for open source statistics on commonly used passwords and characters in passwords to determine password strength.  
Login requires a set password which is stored in a separate file. This may be used to calculate the key for encrypting the other passwords. Both the username and password can be hashed, as they don’t need to be accessed.

Use different keys for login and normal password.

**Research:**

“Identify why your project might be of importance (not just that you are going to be assessed on it!).”

**User stories:**

As a user, I want to have easy access to my passwords so that I can spend less time signing in and won’t lose them.

As a user, I want my log in information to remain secure, so it is unlikely for my accounts to be breached.

As a user, I want secure information (such as my passwords) to be behind a sign in, so malicious actors cannot access them even if they gain access to my computer.

As a user, I want to be able to reset my login password, so if I forget it, I can still access my other passwords.

As a user, I want the application to be easily accessible, preferably without requiring any prerequisites, so it is easier to install and use.

**Design:**

Detailed design refers to the operations needed, data structures used, algorithms, data storage and program flow/sequence.

Style regards the architecture.

Template is for the **design document**.

**Password manager research:**

Most password managers use AES-256 (src: <https://nordpass.com/features/xchacha20-encryption/>)

Chrome:

* List of websites
* Clicking prompts sign in
* Saves username + password (password obfuscated)
* Click to copy / unhide password
* On sign-in, do not need to sign in again
* [Passwords are encrypted](https://support.google.com/chrome/answer/10311524?hl=en-GB#zippy=%2Chow-we-protect-your-data)

Nordpass (#1 apparently):

* [Passwords stored off-device, but local key.](https://nordpass.com/features/zero-knowledge-architecture/)
* [Special encryption algorithm](https://nordpass.com/features/xchacha20-encryption/) (C or C++ by the looks of things)

<https://gist.github.com/jo/8619441> has a list of JS cryptography libraries.  
WebCryptoAPI (<https://www.w3.org/TR/WebCryptoAPI/>) is a common one.

**For limitations:** As we are not allowed to use languages other than HTML, JavaScript and CSS, the number of available cryptographic libraries/APIs are limited. It was resolved to use the crypto API.

**Issues:** I have discovered that file handling is difficult, as JS treats it different to the C languages. Thankfully, JSON files are handled in a different, easier to use, way.  
Unfortunately, to use the standard method of accessing JSON files, a server would be needed to be hosted (otherwise a CORS error occurs). Eventually, I decided to simply ask the user for the storage files, which I would consider a bad implementation if it was possible to implement in any other way. On the upside, this allows for multiple users to store their credentials.

Due to the number of elements needing to be uniquely identified, the function for them cannot use the shorter (and probably faster) template feature of HTML, instead relying on manually creating each element. This is an inefficient implementation, and could be improved through taking a template and modifying IDs.

As with loading files, saving cannot be implemented as desired. Due to the aforementioned CORS errors, when the user wishes to save their data, they must download a new, unique file. As the existing password file is not deleted, this is prone to cause confusion.

Due to the use of promises for the hashing function and my lack of knowledge regarding asynchronous operations, I could not use functions to perform the operations. In addition, the majority of both signIn and updateFile ended up in then blocks.

### FOR VIDEO

Project Vision:  
A web application that stores usernames and passwords in a secure format locally, which can only be accessed by signing in.

Background:  
Other such password managers exist (EXPAND)

Project Plans and Sprints:  
I have done three sprints, performing research on the topic in the first, designs for the application in the second, and prototyping and initial code development in the third.

UML Models:  
*Show class diagram, make more?*

Prototype:  
Show prototype, ideally semi-functional.

Description of challenges / issues faced:  
Cannot use other languages for the security side, meaning the encryption algorithm is exposed.

## Mark Scheme

Interim Practice (Video):

A screenshot of a computer

Description automatically generated  
A grid of text boxes

Description automatically generated with medium confidence  
15%’s are equivalent to 0.9 marks, 20%’s are equivalent to 1.2 and 35’s are 2.1. A reminder that this is only valued at 6 of the 100 marks.

Final Practice (Presentation):  
A white sheet with black text

Description automatically generated  
A white rectangular box with black text

Description automatically generated with medium confidence

**Final Portfolio:**  
A white sheet with black text

Description automatically generated with medium confidenceA white rectangular box with black text

Description automatically generated with medium confidenceA white grid with black text

Description automatically generated with medium confidenceA white rectangular grid with black text

Description automatically generated