Password Manager

(Title WIP)

# Introduction

This report presents the COMP1004 coursework. The password manager created for the coursework focuses on security and data security.

Alongside the coursework, the Software Development Lifecycle (SDLC) is also discussed. An agile driven approach has been used for this project.

# Use of the Software Development Lifecycle

(Name WIP and section pending)

This project was planned over a week, with other password managers (such as the chrome password manager) being investigated.

Once planning concluded, the rough period of development was established, with sprints being assigned tentative tasks.  
A screenshot of a computer

Description automatically generated

*Very basic, terrible compared to template.*

# Design

(Design pending)

## Project Vision

This password manager is for individuals to help store their passwords in a secure way. The application stores passwords in encrypted files, and these are only decrypted when a successful sign-in occurs, preventing unauthorised access to passwords.

## Background

Password managers are important for the secure, everyday use of computers. Provided they are competent in securing their accounts, a person will have several passwords, most of which are not memorable. A user can either store their passwords physically (which is less portable), in an unsafe plaintext file, or in a secure password manager.

Password managers are effectively databases which store usernames and passwords for different sites in a controlled form. A strong manager protected behind a strong access password means the user only needs to remember one password (Li, 2014)

A password manager must be secure, otherwise it can introduce a significant vulnerability by potentially exposing all of a users’ passwords. Many managers have security flaws, with 4 out of 5 studied password managers in a study being vulnerable to attacks which reveal credentials (Li, 2014).

Regardless of where password manager data is stored: portable, cloud or local, there are dangers. This is because USB Memory Sticks and Smartphones can be stolen, and cloud storage can be accessed without permission by attackers. Though storing locally is more secure, requiring an attacker to either gain access to the computer, it is still not acceptable. Because of these threats, the password manager must use a secure format. (Gasti, 2012).

## User Stories and Use Case Scenarios

**This is heavily WIP. Reformat and add introduction.**See Template for a correctly done section.

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.  
Note: As this is a web-based project, the JavaScript is visible to anyone using inspect element. For security reasons it is necessary to find a way of hiding the encryption method. The best way of achieving this is through an API.

## Architecture

(Architecture pending – the architecture contains UML class diagrams)

(Note: There is a sitemap in the template, but this is a SPA, so it is irrelevant)

## Page Design

(Also known as “Wireframes” in the template)

A black text on a white background

Description automatically generated

A sign in box with red text

Description automatically generated

The sign-in box is displayed on top of the main password manager page, which does not yet have any information displayed, and may be blurred.

Warning: A wireframe is a webpage design with javascript elements and images hidden.

A screenshot of a computer

Description automatically generated  
This is an initial prototype page.  
When functional, no passwords will be shown until the user signs in, at which time they will be decrypted and the containers filled.  
The contents of Website 1 can be shown and hidden.  
Show Password sets the password box from the “password” type to “text”, making it visible and able to be directly copied.  
Copy Password copies the contents of the password box to clipboard without showing it.

## Issues and Constraints

(Pending, discusses issues of learning new technologies, planning and accounting for modules and deliverables)

This project is limited in terms of what can be used. Ideally, an encryption algorithm would be in C#, C or C++, unfortunately it is required that the scripting side of the project is exclusively done in JavaScript. This means that the application is less secure than alternatives, and many open-source encryption / hashing algorithms cannot be used.

# GitHub Repository

<https://github.com/CentralC0re/COMP1004>

# References

All references here. Standard format:  
*Title, Section (page, chapter, etc), Source: URL (Access Date)*

The Emperor’s New Password Manager: Security Analysis of Web-based Password Managers (Zhiwei Li et al), Page 3, Section 2.1, Source: <https://www.usenix.org/system/files/conference/usenixsecurity14/sec14-paper-li-zhiwei.pdf> (Accessed 28/11/23)

On The Security of Password Manager Database Formats (Paolo Gasti and Kasper B. Rasmussen), Section 1, Source: <https://ora.ox.ac.uk/objects/uuid:926086ea-180b-4f11-a599-2522a80837f4/download_file?file_format=application%2Fpdf&safe_filename=pwvault.pdf&type_of_work=Conference+item> (Accessed 28/11/23)