

Analysis

Project Vision and Background

This report presents the plan for my COMP1004 module project, which focuses on creating a single-page application (SPA) aimed at managing Passwords and saving them for different accounts. This project is intended to help users track and manage their passwords.

The report will outline the project vision and UML diagrams along with Implementation Process. The Report will also discuss the challenges faced, the planning process, and how I approached the problems faced.

The Password Manager application addresses the challenges of managing and securing online accounts by offering a centralized, secure solution for password storage. With the increasing number of services requiring unique passwords, many users resort to weak or reused passwords, compromising security. This application provides encrypted storage and encourages people to use strong passwords when they are creating their account, allowing users to securely store, manage, and retrieve their passwords.

Key features include User authentication with registration and login, encrypted password storage, a user-friendly interface for adding, viewing, editing, and deleting passwords and password categorization and search functionality for easy access, backup and recovery options to protect user data.

Project targets individuals seeking to improve their online security, the project aims to make password management both secure and convenient by ensuring compliance with industry-standard encryption. Password Manager helps safeguard online identities and sensitive information. Feedback from staff and peers will refine the project to meet security and usability standards.

SDLC (Software Development Life Cycle)

The project will be executed in an agile environment, where iterative development is encouraged to allow frequent reviews, improvements, and adaptability to changing requirements.

The requirements and features will be outlined based on research, feedback, and observations.

User interface (UI) and system architecture design will be done.

Development will occur in iterations, with each sprint focusing on a specific feature or broader.

Frequent testing will be conducted to ensure that the features are functional and secure in every iteration will be recorded.

Once the final version is stable, the application will be deployed on github.

Ongoing support and updates will be provided to improve performance, security, and user experience.

Research plan

The means of collecting data I have chosen are write a questionnaire to be completed by the teaching staff and Students also I will be doing 2-week sprint meetings with my tutor, lastly, to conduct an interview with my possible end users it may not be the best way of collecting data but it will be enough for what we are doing.

Observations related to research plan

I chose to observe the way the Websites are done in real world environment as it can give me a better understanding of what features I feel may be suitable to include in a new system and it may help me see what aspects of the work seem most confusing to us the students and ways I could go about designing a system that is suitable for specifications also understanding these specification is useful to me as I can see how the technologies being limited in some way that would affect the development cycle

Questionnaire & Interview

Questionnaire

Q1-) How should the user interface needs to be like ?

Q2-) I intend to implement Login/Register System do you think this is a good idea ?

Q3-) I also Intend to add encryption protect user data what do you think about this ?

Q4-) What else do you think we can implement apart from the specifications of this project ?

Q5-) Lastly How should I Approach the development of this website ?

Interview

Jamey

How should the user interface needs to be like ?

- User Interface needs to be clear and easy to navigate between pages.
- There shouldn't be any complicated menus to make this clear as possible

I intend to implement Login/Register System do you think this is a good idea ?

- Yes, many industry professionals use login and register to create accounts to associate the account information to a specific user also it protects user data in some degree.

I also Intend to add encryption protect user data what do you think about this ?

- In accordance with UK user data protection laws, I think this is a really good idea this would make the website secure as any it protects the website from any ill intended attack to steal user data or something else.

What else do you think we can implement apart from the specifications of this project ?

- You could add when people registering adding randomly generated passwords which has rules that where it's going to be a strong password improving security.

Lastly How should I Approach the development of this website ?

- I believe you should use agile methodology as this project will be complicated and you may need to follow this methodology to update the project any point in time

Teacher

How should the user interface needs to be like ?

- User interface needs to be easily navigable

I intend to implement Login/Register System do you think this is a good idea ?

- Yes, Login/Register System is essential to almost every website.

I also Intend to add encryption protect user data what do you think about this ?

- This is great idea but with the technology restrictions you may struggle to achieve this feature

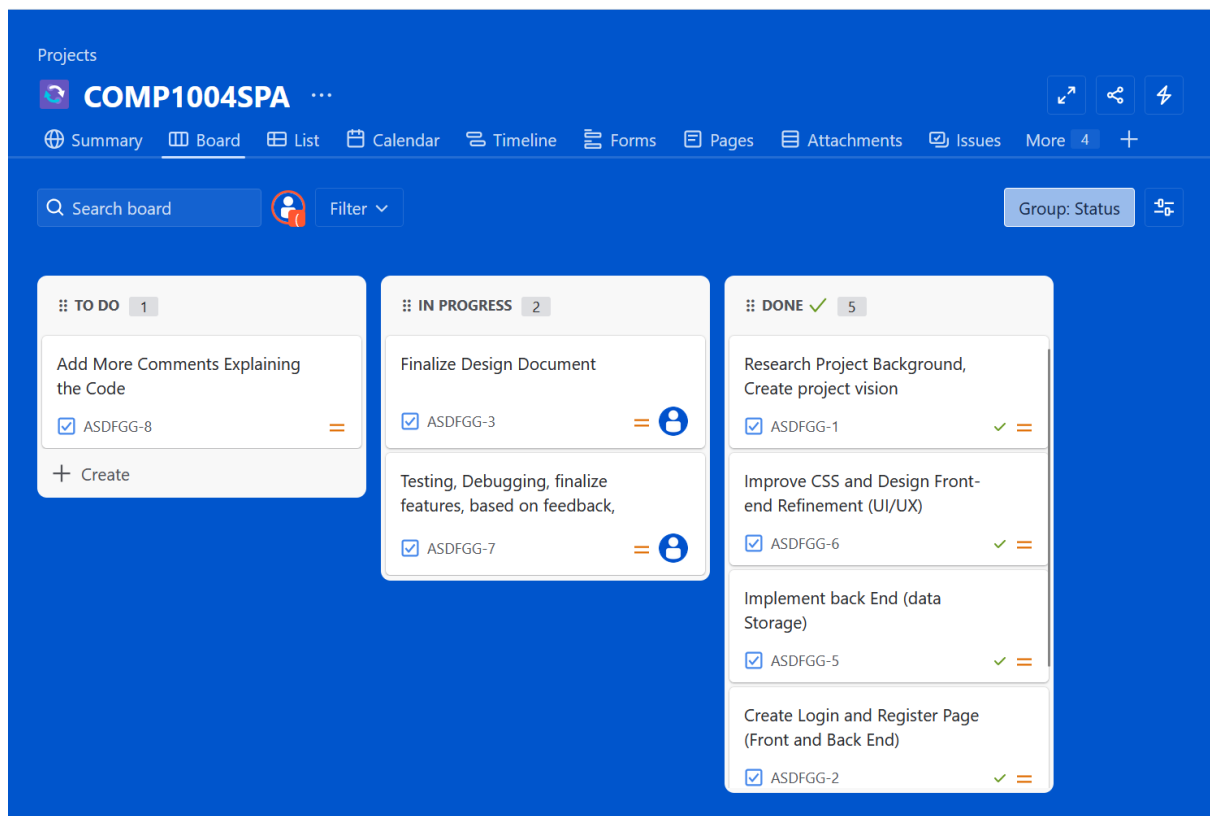
What else do you think we can implement apart from the specifications of this project ?

- I have nothing else to add apart from make sure to not go out of coursework restrictions also use plenty of comments to make code readable to you and others alike.

Lastly How should I Approach the development of this website ?

- Ther is loads of methodologies you can use but, in this module, we will use agile methodology as this is the primary methodology used by a lot of industry professionals

Sprints



Iteration

TO DO0

What needs to be done?

☒

▼

IN PROGRESS2

Password Genaration

☒

ASDFGG-10

Encryption

☒

ASDFGG-9

DONE✔8

Research Project Background, Create project vision

☒

ASDFGG-1

Finalize Design Document

☒

ASDFGG-3

Testing, Debugging, finalize features, based on feedback,

☒

ASDFGG-7

Tasks

Items	October – December '24	January – March	
<input checked="" type="checkbox"/> ASDFGG-1 Research Project Background, Create project v	<div></div>		
<input checked="" type="checkbox"/> ASDFGG-2 Create Login and Register Page (Front and Bac		<div></div>	
<input checked="" type="checkbox"/> ASDFGG-3 Finalize Design Document		<div></div>	
<input checked="" type="checkbox"/> ASDFGG-4 Create Password Management App page		<div></div>	
<input checked="" type="checkbox"/> ASDFGG-5 Implement back End (data Storage)		<div></div>	
<input checked="" type="checkbox"/> ASDFGG-6 Improve CSS and Design Front-end Refinemer	<div></div>		
<input checked="" type="checkbox"/> ASDFGG-7 Testing, Debugging, finalize features, based or		<div></div>	
<input checked="" type="checkbox"/> ASDFGG-8 Add More Comments Explaining the Code		<div></div>	

Pending Tasks

Items	October – December '24	January – March	
<input checked="" type="checkbox"/> ASDFGG-3 Finalize Design Document		<div></div>	
<input checked="" type="checkbox"/> ASDFGG-7 Testing, Debugging, finalize features, based or		<div></div>	
<input checked="" type="checkbox"/> ASDFGG-8 Add More Comments Explaining the Code		<div></div>	

Last Iteration

Items	December '24	January – March	April – June
<input checked="" type="checkbox"/> ASDFGG-1 Research Project Background, Create project v	<div></div>		
<input checked="" type="checkbox"/> ASDFGG-3 Finalize Design Document		<div></div>	
<input checked="" type="checkbox"/> ASDFGG-7 Testing, Debugging, finalize features, based on		<div></div>	
<input checked="" type="checkbox"/> ASDFGG-8 Add More Comments Explaining the Code		<div></div>	
<input checked="" type="checkbox"/> ASDFGG-2 Create Login and Register Page (Front and Back)		<div></div>	
<input checked="" type="checkbox"/> ASDFGG-4 Create Password Management App page		<div></div>	
<input checked="" type="checkbox"/> ASDFGG-5 Implement back End (data Storage)		<div></div>	
<input checked="" type="checkbox"/> ASDFGG-6 Improve CSS and Design Front-end Refinement	<div></div>		

Cancelled Tasks

Items	December '24	January – March	April – June
<input checked="" type="checkbox"/> ASDFGG-10 Password Generation			
<input checked="" type="checkbox"/> ASDFGG-9 Encryption			

Progression

ASDFG	Deliverable	Sprint 1	Sprint 2	Sprint 3	Sprint 4	Sprint 5	Sprint 6	Sprint 7	Sprint 8	
1	Research, Project Background	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>
2	Design Document	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>
3	Testing, Debugging	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>
4	More Comments	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>
5	Login And Register Page	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>
6	Password Management App page	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>
7	Implement Back End	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>
8	Improve CSS	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>
9	Password Generator	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>
10	Encryption	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>

<div></div>	Not Started
<div></div>	Cancelled
<div></div>	Partially Implemented
<div></div>	Implemented Untested
<div></div>	Testing
<div></div>	Tested
<div></div>	Ready

Finalised Requirements and Evaluation

Based on the feedback received from the questionnaire and interviews, I have outlined the following finalized requirements

Registration and login functionality with secure data handling is a must as every user needs to have their own unique way of seeing their data its also protects the user from getting their passwords stolen Implementation of strong encryption for storing user passwords is also another point as it's a must have as Laws for protecting the data of the users is strict.

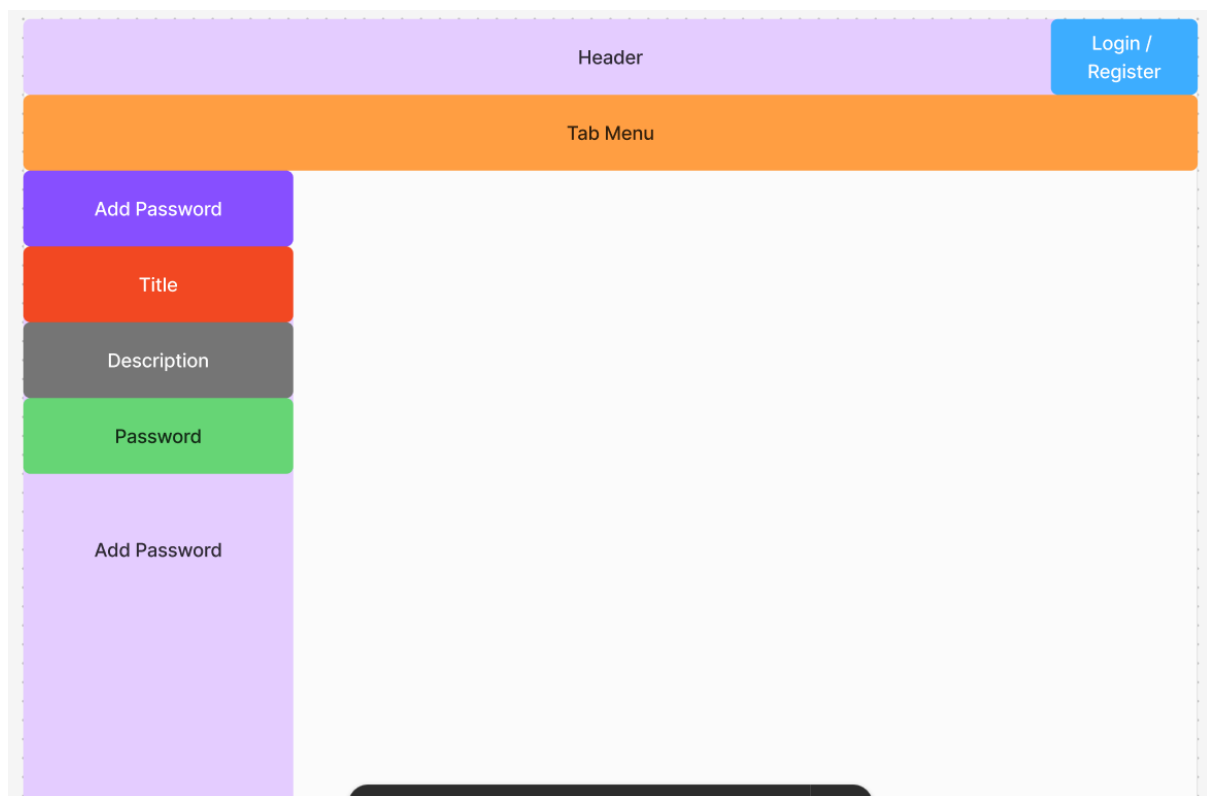
A clear, intuitive interface for managing passwords is a must have as well as badly designed website will create frustration for user experience.

A password strength measuring for creating strong passwords will be included.

Lastly the project will be developed using an agile methodology to allow for iterative updates and flexibility in meeting goals.

Design

Initial Designs



Main Menu

Main Menu consists of a header showing name of the project

A button for generating Password where you input the title description and password where then it generates a block where it has the information you have inputted

Tab Menu is the typical form of navigation used by most programs. It involves an almost tab type navigation but instead of a drop-down menu I chosen a static one

Advantages:

- Easy to use
- Simple

- Cost Effective (Even though this website won't cost me anything I put this as in real world applications it matters)
- Time Efficient (Development Time will be less without the complicated features).

Disadvantages:

- Finding what you're after can be difficult as the user won't know where each option will be until they've used it a few times
- No Way to Delete passwords apart from deleting JSON file.

Improvement On the Design based on Feedback



Because our website is very simple I removed the Tab menu as there wasn't much to it instead, I put a description.

Login Page

Header

Login / Register

Description of The Page

Username or Email

Password

Login

Register

Essentials like header Description, Login / Register haven't changed

But I added specific entry forms for registering and login on

Additionally, you can jump for between register and logging page by clicking Register and Login on the bottom left corner button which is Gray

No Improvements made its good as it is from feedback

Register

But I added specific entry forms for registering and login on

Additionally, you can jump for between register and logging page by clicking Register and Login on the bottom left corner button which is Gray

No Improvements made its good as it is from feedback

Final Version

Password Manager

[Login/Register](#)

Login

Don't have an account? [Register here](#)

Password Manager

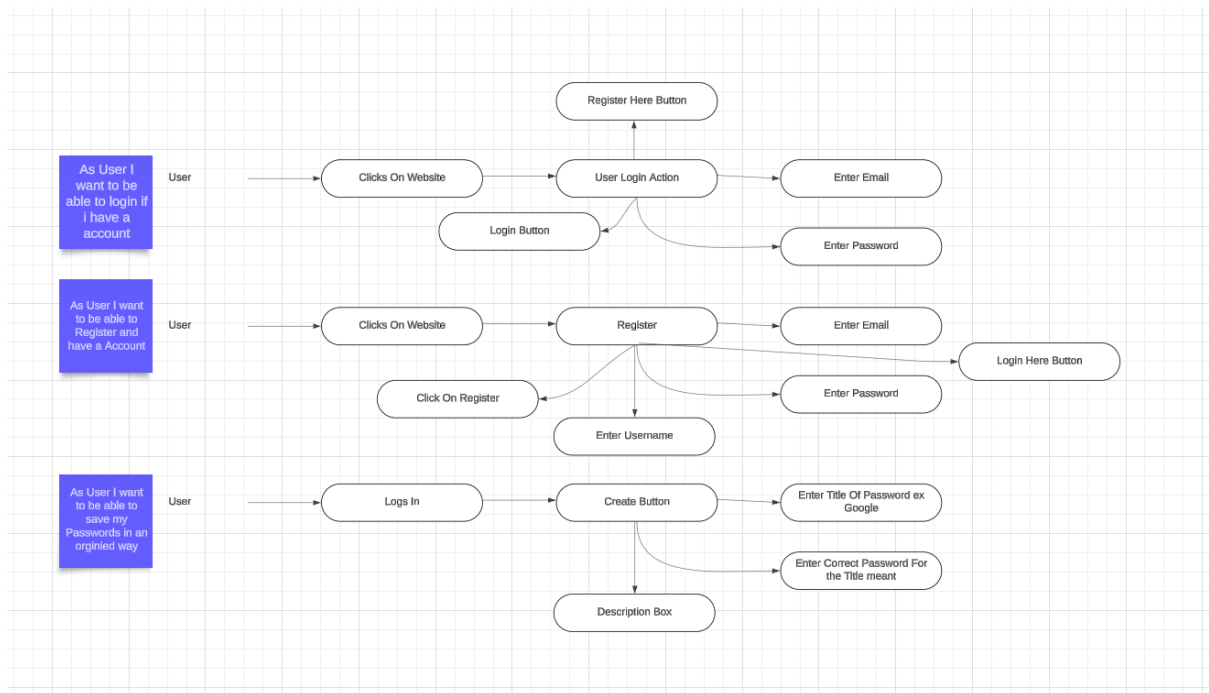
[Login/Register](#)

Register

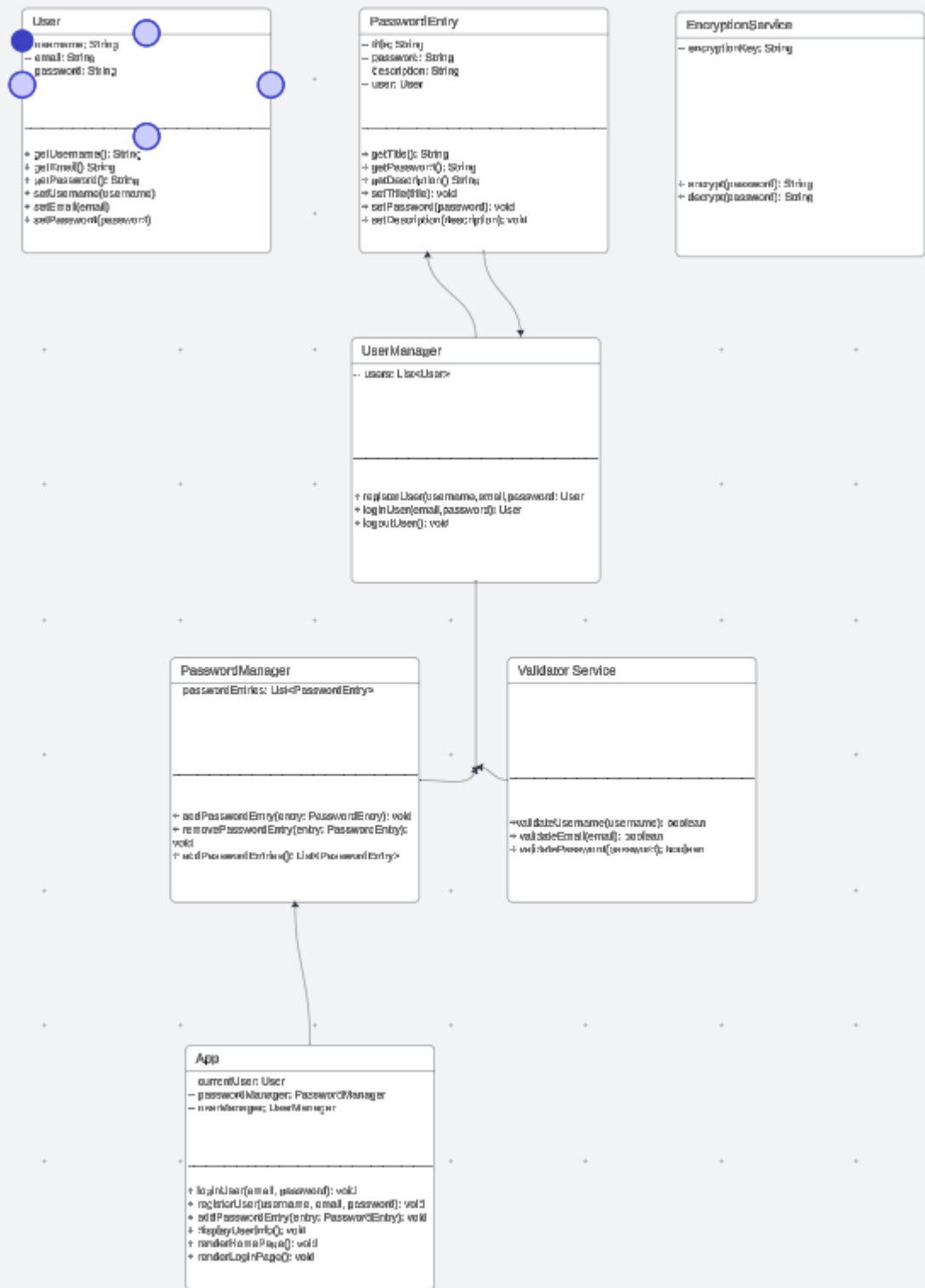
Already have an account? [Login here](#)

UML Diagrams

Use Case Diagram



Class Diagram



Class Diagram Overview:

User class: Contains user-specific data such as email, username, and password.

Password Entry class: Contains details about individual passwords, including title, password, and description.

Encryption Service class: Handles encrypting and decrypting passwords to ensure sensitive data is secure.

User Manager class: Manages user-related operations like registration, authentication, and password entry management.

Password Manager class: Manages password entries, storing and retrieving them for a particular user.

Validator Service class: Ensures the correctness and security of inputs such as usernames, emails, and passwords.

App class: Manages the overall application workflow (e.g., logging in a user, rendering pages, etc.).

Breakdown of Interactions:

User -> Password Entry:

A **User** can have multiple **Password Entry** objects. This relationship is represented as a one-to-many relationship (one user, many password entries).

User Manager: **User Manager** is responsible for handling **User** objects (e.g., creating, authenticating, and logging out users).

It interacts with the **User** class to authenticate and register users.

Password Manager: **Password Manager** is responsible for managing the password entries for each user. It stores multiple **Password Entry** objects and provides functions like adding, removing, and retrieving entries.

Password Manager interacts with **User Manager** to ensure that password entries are associated with the correct user.

Encryption Service: **Encryption Service** provides security for the password entries by encrypting the password before storing it and decrypting it when retrieving it.

The **Password Entry** objects call the **Encryption Service** when a password is added or retrieved, ensuring passwords are never stored in plaintext.

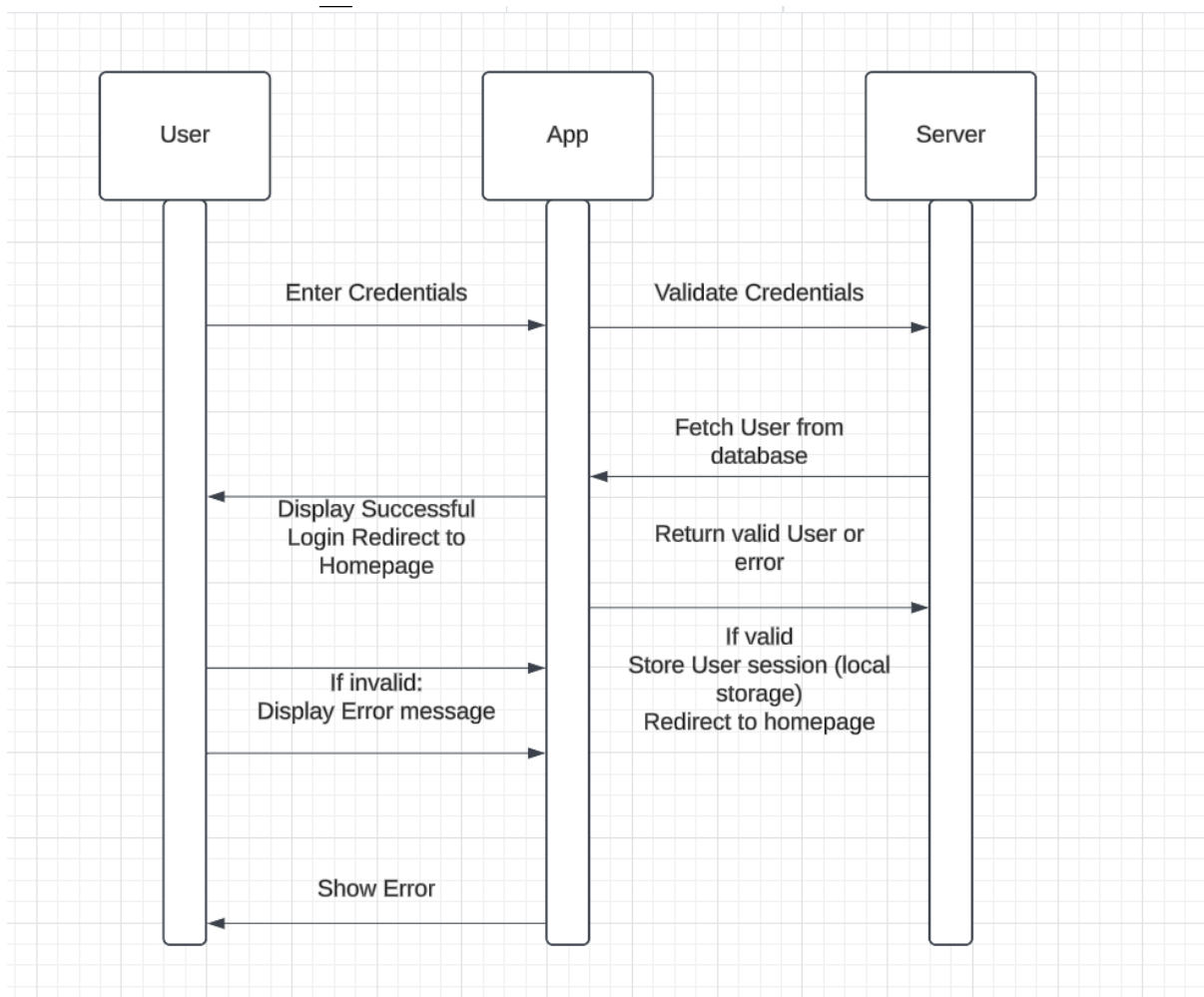
Validator Service: **Validator Service** is responsible for validating user input, such as checking the validity of a **username**, **email**, and **password** (e.g., ensuring strong password rules or proper email format).

User Manager uses **Validator Service** to ensure inputs are valid when users register or update their profile.

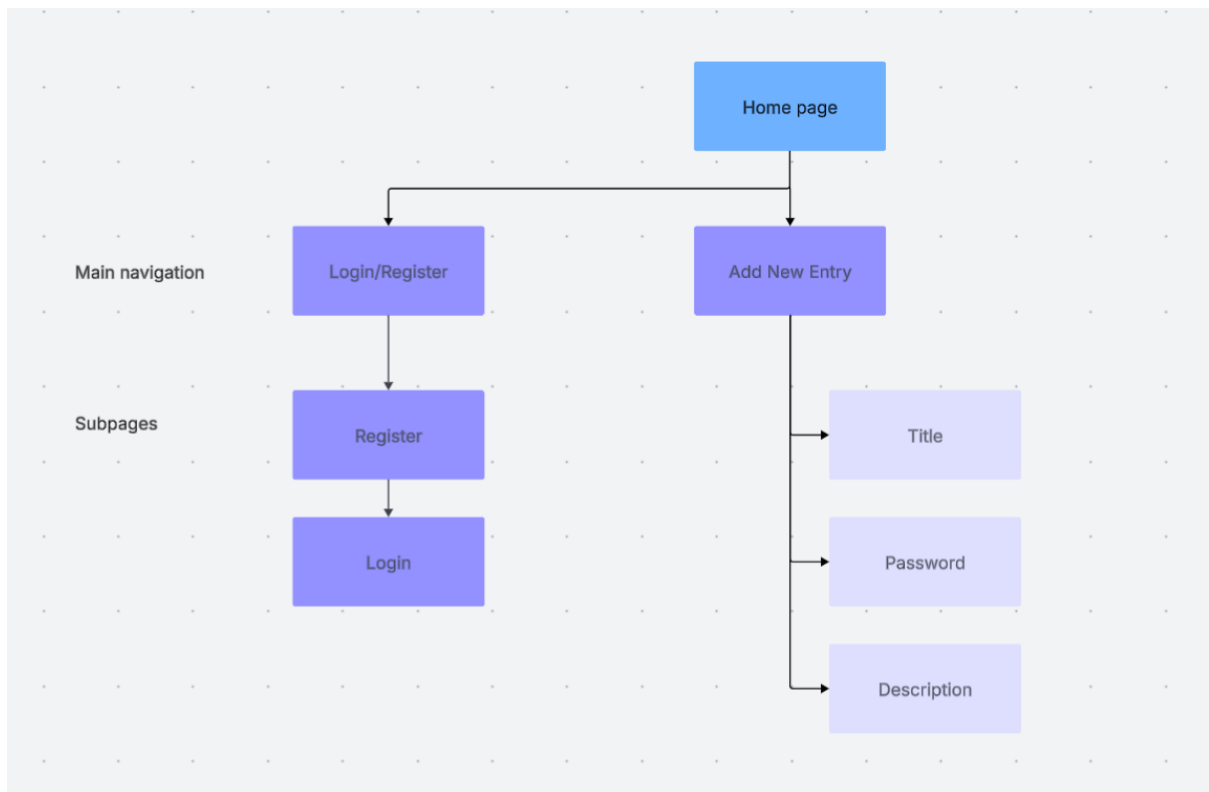
App: **App** is the central controller of the application and interacts with the **User Manager**, **Password Manager**, and **Encryption Service** to manage the overall application flow.

The **App** class can log in users, register new users, add new password entries, and render various pages based on user interactions.

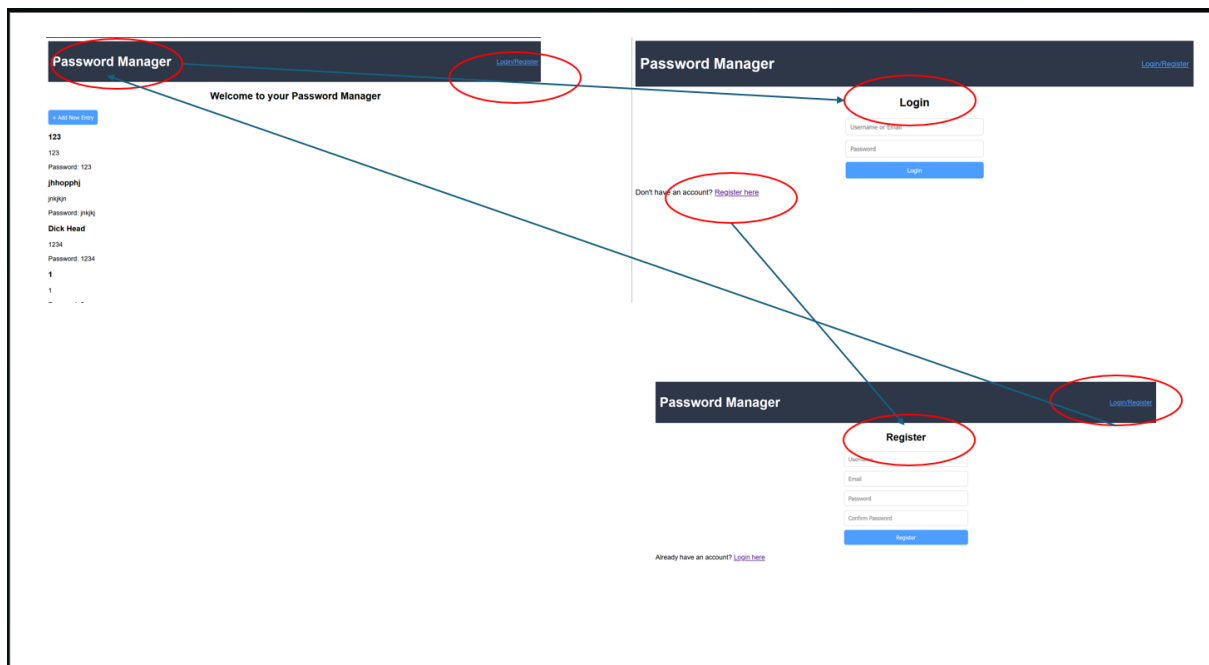
Sequence Diagram



Sitemap



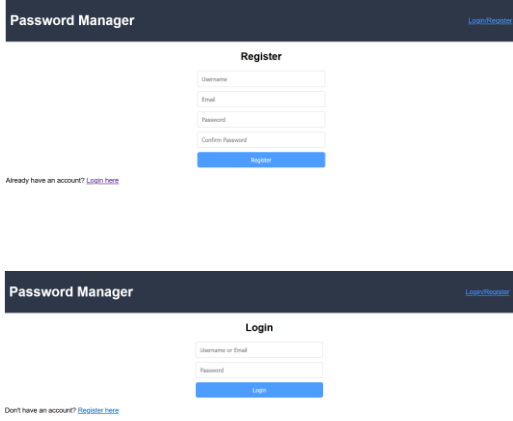
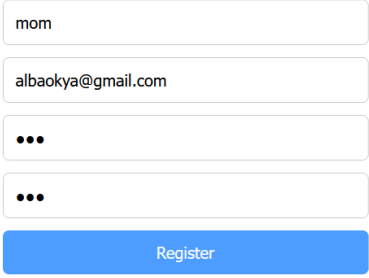
Wireframes

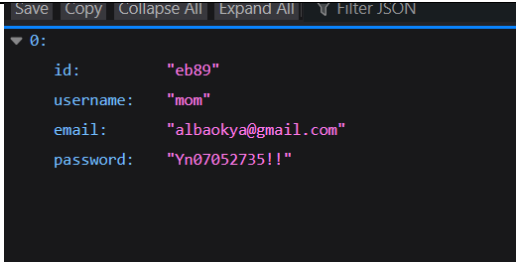
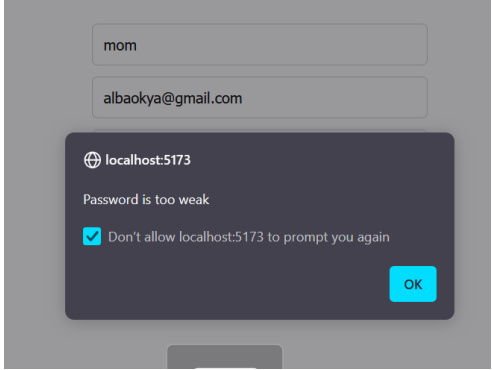
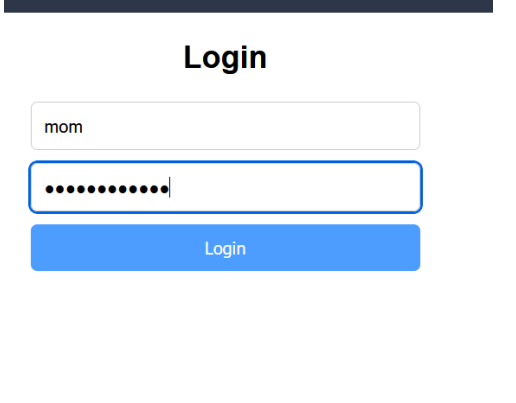
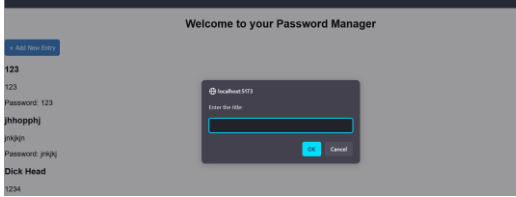



Testing And Implementation

Development Testing

The testing phase is to ensure that the key features of the Password Manager worked as expected and met to the project specifications. Throughout the development process, I conducted several rounds of testing across different areas of the application to ensure functionality and usability.

What is Getting Tested	How to do the Test	Result	Evidence	Comments
Login/Register page	Open the Login and Register Page	Pass		Result may be a pass, but this process is a bit login I suspect its due to routing that I have done
Registering	Registering via email and password	Pass		Registering is successful application also records the registered account into a json file

				
Password Strength	Create an inappropriate password and try to register the account	Pass		Due to password being too weak and password not including good practices application asks users to put password
Login	Login via created username or email	Fail		
Add New Entry Button	Enter Required field to record a password	Pass	 <p>Google</p> <p>Description</p> <p>Password: Yn07052735</p>	Password appears in the page where user can see it but another user can as well because we can't really login

Password record	Check storage of the files for passwords	Pass	 <pre> JSON Raw Data Headers Save Copy Collapse All Expand All Filter JSON password: "1" description: "3" user: null 6: id: "dc07" title: "1" password: "3" description: "3" user: null 7: id: "c24e" title: "fddfsd" password: "dsff" description: "fd" user: null 8: id: "9349" title: "sdfsdfs" password: "sdf" description: "df" user: null 9: id: "3f99" title: "Jamey" password: "is" description: "gay" user: null 10: id: "e169" title: "Google" password: "Yn07052735" description: "Description" user: null </pre>	As can be seen in the evidence no password is assigned to any user resulting in everyone being able to see passwords

Key Observations & Issues Identified During Testing:

Login/Authentication Failures:

There was an issue with logging in users that was linked to routing issues. The system was not properly managing user sessions, resulting in login failures. There was no proper error messages displayed to the user when login failed.

Password Storage (Plaintext Issue):

Passwords were stored in plain text in the JSON file without encryption, posing a significant security risk.

Password Strength Validation:

The password strength validation worked well, rejecting weak passwords. This was a necessary feature for improving security.

Password Entry Visibility:

A significant security flaw was that once passwords were entered into the system, they were visible to all users who accessed the page.

Routing & Navigation:

During testing, some inconsistencies in navigation were discovered, especially between the login and registration pages.

UI and UX:

While the interface was simple and functional, the design could benefit from additional user feedback and more intuitive navigation elements.

Future Planned Improvements: Implement session management, add error handling for incorrect login attempts, and provide user feedback for better UX.

Introduce encryption techniques hashing for password storage, ensuring user data is protected.

Implement more detailed feedback for users, explaining why their password was considered weak and how to strengthen it.

Implement a user authentication system that ensures password entries are only visible to the registered user, implementing role-based access control for user data.

Refine routing and add clear visual cues for transitions between login and registration screens to improve user experience.

Implement more user-friendly navigation and visual elements that guide the user through the application, reducing friction in interactions.

Ongoing Improvements and Future Testing

Future iterations should focus on adding encryption for password storage using techniques such as AES (Advanced Encryption Standard) or bcrypt to hash passwords.

This will improve security by ensuring that passwords are not stored in plain text but unfortunately technology restrictions prevent us from ever implementing this.

Performance and Load Testing:

It's important to conduct performance testing, particularly when handling multiple password entries and user interactions. Load testing can help identify any performance issues or usability issues when the application scales.

Issues And Challenges Faced

In the development of this password Manager website, several constraints and limitations that became apparent that significantly affected the project's design and implementation.

These challenges were caused in both the technological restrictions imposed on the project and my personal time limitations, which increased the difficulty of implementing features, particularly in the context of cybersecurity practices.

Technological Constraints

One of the most significant constraints was the limitation to only use Java, HTML, and CSS. Initially, I had a bigger vision for the website, considering the use of TypeScript, React and server-side languages like Python and extensions for encryption and password generation.

However, the restriction to only Java, HTML, and CSS created challenges.

For example, encryption an important feature for securing stored password could not be properly implemented within the boundaries of these technologies.

Without encryption, the security of the password manager was compromised, as passwords would have been stored in plain text, making the system vulnerable to attacks.

Also, These Restrictions prevented us from using any frameworks that could have helped speed up development and provided stronger security features which most industry professional's use.

While HTML and CSS were sufficient for the basic layout and styling, they were inadequate for the backend functionality required to securely store and user information.

Without access to proper encryption algorithms or secure hash functions, it became clear that the website would not be able to meet the security standards expected from a modern password manager.

Time Constraints and Project Adjustments

I had to make several difficult decisions regarding which features were feasible to implement in the given time. I underestimated the required time to implement essential features within the limited technological scope. Some features, such as encrypted password storage had to be removed from the project due to time limitations.

Additionally, my personal time constraints played a significant role in the project's progress. With a part-time job requiring me to work 20 hours a week, spread across three days, my availability to leave time to the project was limited.

Balancing work and academic responsibilities made it challenging to meet project sprints and deadlines, which ultimately delayed the implementation of some features.

I spoke with some of my peers, and many of them also had to revisit or redo parts of their projects. I have concluded that they had misunderstood the technology restrictions as did I. The confusion surrounding what technologies were allowed also added to the projects missing key features. According to my interview previous year's students had used a wider range of technologies during their sprint meetings, leading most students (me being included) to assume they could integrate those same tools. Unfortunately, this assumption was incorrect, which caused confusion and further delayed progress. This miscommunication highlighted the need for clearer project guidelines.

Cybersecurity and Legal Issues

Due to the technological constraints, it was not possible to implement some of the cybersecurity industry practices that would have been good for a password manager. This created a situation where, although the website could store passwords, it did so in

an insecure manner, which violated key cybersecurity principles around data protection and user privacy.

A good cybersecurity framework would typically involve using encryption to secure passwords, implementing secure communication protocols like HTTPS, and ensuring proper user authentication.

Without these security features, the project was not able to meet the expectations of users typically expected of a password manager.

From a legal perspective, this posed issues as well, particularly concerning data protection laws like GDPR.

A password manager is inherently sensitive and requires careful handling of user data

Evaluation

The development of the password manager website faced a lot of challenges that affected its functionality, security, and performance. These challenges were primarily caused by technological constraints, time limitations, and a lack of clear communication regarding project requirements. The result was a working prototype, but one with significant security flaws and missing key features necessary for a functional, secure password manager.

The restriction to using only Java, HTML, and CSS was the most significant constraint. While these technologies are sufficient for creating basic web pages, they lack features for securely managing sensitive user data.

The project suffered from tight deadlines and the personal time limitations of balancing academic responsibilities with a part-time job. This resulted in the need to prioritize basic functionality over more critical features.

Miscommunication regarding the allowed technologies created confusion about the scope of the project.

The most significant issue with the project was the lack of proper encryption and secure storage mechanisms for passwords.

Conclusion:

While the password manager website succeeded in fulfilling basic requirements such as user registration and password entry, it ultimately fell short of meeting the advanced features for security and functionality that are essential for a password manager application.

The main issues were secure password storage, encryption, and proper user authentication due to technological and time constraints.

From this experience, it became clear that clear communication regarding project requirements, proper time management, and a strong focus on data protection are essential for building secure and functional software. Although the project met expected requirements and tests it fallen short on advanced features, this showcases the importance of including encryption, secure data storage, and user authentication mechanisms in any password management system.

To improve the project, future work should focus on implementing proper encryption, secure storage, and authentication. Additionally, more flexibility in the allowed technologies and clearer project guidelines would help to prevent misunderstandings and ensure that projects are developed to their full potential.

References

<https://en.wikipedia.org/wiki/Bcrypt>

<https://ico.org.uk/for-organisations/data-protection-and-the-eu/data-protection-and-the-eu-in-detail/the-uk-gdpr/>

https://en.wikipedia.org/wiki/General_Data_Protection_Regulation

Note: EU GDPR and UK GDPR technically different but almost the same

<https://www.w3schools.com/java/default.asp>

<https://www.w3schools.com/css/default.asp>

<https://www.w3schools.com/html/default.asp>

<https://www.w3schools.com/react/default.asp>

https://www.w3schools.com/bootstrap/bootstrap_ver.asp

Note bootstrap and react has been removed

<https://www.w3schools.com/howto/default.asp>

<https://www.ocr.org.uk/Images/269250-f454-exemplar-candidate-work-project-a-high-band-.pdf>

<https://www.ocr.org.uk/Images/77802-unit-f454-exemplar-candidate-work.pdf>

https://www.pgonline.co.uk/library/seeinside/9781910523193_ocr_projects.pdf

<https://www.npmjs.com/package/concurrently>

<https://www.npmjs.com/package/json-server>

<https://github.com/dropbox/zxcvbn>