1. Declaration

I, Hao Li, declare that this assignment, titled [Assignment Title], is my own original work and has not been copied from any other source except where explicitly acknowledged. I have not engaged in plagiarism, collusion, or any other form of academic misconduct in the preparation and submission of this assignment. All sources of information and data used in this assignment have been properly cited and referenced in accordance with the prescribed guidelines. I have not used unauthorized assistance in the preparation of this assignment and have not allowed any other student to copy my work. I am aware that any breach of academic integrity may result in disciplinary action as per the policies of Monash University, which may include failing this assignment or the course, and further academic penalties.

Signature: Hao Li   Date: 2025.7.27

2. Github Check

Enter your Github details here.

Github Username *Enter your username here*	HaoLi104
A2 Shared? *Have you started and shared your assignment repository with your tutor yet?*	*HaoLi104/FIT5032*

3. Self-Evaluation

Rate your performance for each criteria. Put a ✅(tick) in the box where you think your work belongs.

Criteria	Exceeds Expectations	Meets Expectations	Needs Improvement	Fail to meet expectations
BR (A.1): Development Stack and Coding	✅			
BR (A.2): Responsiveness	✅			
BR (B.1): Validations	✅			
BR (B.2): Dynamic Data & Data Structure	✅			

BR (C.1): Authentication	✅			
BR (C.2): Role-based authentication	✅			
BR (C.3): Rating	✅			
BR (C.4): Security	✅			

4. Screen Recording of BRs

Create a 3 minute video showing your basic web application in action! Upload this video to your Google Drive and put the link here (ensuring that you have updated the access list so its not private).

FIT5032-Video - Google Drive

5. Reflections: Implementation of C.4 Security

If you have implemented BR C.4, in less than 200 words describe the approach that you have taken to implementing Security in your application. What security flaws were you trying to prevent and what security measures have you implemented to fix those flaws? How do you know that these measures will help prevent those issues from happening? Optionally you can cite external sources to provide evidence for your claim.

My application implements a multi-layered security approach to prevent common web vulnerabilities, focusing on **Cross-Site Scripting (XSS)** and **unauthorized access**.

To mitigate XSS attacks via malicious script injection, we exclusively utilize Vue.js's default {{ }} text interpolation, which automatically HTML-escapes all user-generated content. This effectiveness is supported by Vue's official documentation.

For preventing unauthorized access and data tampering, we implemented a robust **role-based access control (RBAC)** system using Vue Router's navigation guards. These guards check route meta fields to protect specific pages (e.g., ensuring only a 'Caregiver' can access /for-caregivers). This blocks unauthorized requests at the routing level before the component is rendered. Additionally, strict client-side input validation, with regex for emails and password strength checks, is enforced to prevent malicious data submission. Secure session management via localStorage persists the authentication state.

6. Reflections: Challenges

What has been the most challenging part of this assignment for you? How has this stretched you as a programmer?

Primary Challenge: State Management and Authentication Flow

The most challenging part of this assignment was implementing a robust authentication system with proper state management using Pinia. Coordinating between localStorage persistence, Vue Router navigation guards, and reactive state updates across multiple components proved to be a complex orchestration. Initially, I struggled with maintaining authentication state consistency when users navigated between pages or refreshed the browser. The challenge was ensuring that the Pinia store, localStorage, and router guards all remained synchronized while providing a seamless user experience.

Technical Complexity: Role-based Access Control

Implementing role-based authentication with Vue Router's navigation guards required deep understanding of both the router lifecycle and state management patterns. Creating a system where different user roles could access specific pages while maintaining security was particularly challenging. I had to carefully design the meta fields in routes and implement proper error handling for unauthorized access attempts.

How This Stretched Me as a Programmer:

Architecture Thinking: This project forced me to think about application architecture holistically, considering how different components interact and share state. I learned to design systems that are both secure and maintainable.

Security Mindset: Implementing security measures taught me to think like a potential attacker, considering various attack vectors and implementing appropriate defenses. This security-first approach is crucial for real-world applications.

State Management Complexity: Working with Pinia and understanding reactive state management helped me appreciate the importance of predictable state flows in modern web applications.

Component Design: Creating reusable components like AggregateRating that maintain their own state while integrating with the global authentication system improved my component design skills.

Error Handling: Implementing comprehensive validation and error handling across forms and authentication flows enhanced my understanding of user experience and defensive programming practices.

This assignment significantly improved my ability to build complex, production-ready web applications with proper security considerations and user experience design.

7. Declaration: Additional Help

Any tools that you used (including Gen AI or existing code reuse) must be declared here.

Note: GenAI is not allowed for coding purposes in any assignment,

However, you may use GenAI for brainstorming and problem solving. You need to declare all such uses here. One row per help used.

Name	Description
Gemini	*I used Gemini for problem-solving when writing my "Reflections" sections. My initial drafts were over the word limit and not well-structured. I provided my original points, and the AI helped me to rephrase and condense them into a more concise and coherent paragraph, allowing me to meet the assignment's formatting and length requirements without changing my core ideas.*