1. Declaration

I, Lau Yiu Ho, declare that this assignment, titled A3: Advanced Web App, 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: _Suey_____ Date: _____2024-10-10_____

2. Github Check

Enter your Github details here.

Github Username *Enter your username here*	suey147
A3 Shared? *Have you started and shared your assignment repository with your tutor yet?*	https://github.com/suey147/FIT5032-Assignment

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 (D.1): External Authentication	• Firebase Auth for secure • User-friendly • Intuitive login flow ○ Error handling			
BR (D.2): Email	• Send with Gmail API • Send with Attachment - receipt of registered event • Smooth user experience for email composition and delivery. • Simple interface			

Criteria	Exceeds Expectations	Meets Expectations	Needs Improvement	Fail to meet expectations
	provide feedback on sending … send			
BR (C.3): Interactive Table Data	• Two tables: event and knowledge hub table • Features: sorting, searching, pagination • Can be filtered by individual columns			
BR (D.4): Deployment to the Cloud	• CloudFlare			
BR (E.1): Cloud Functions	• Serverless cloud function • Benefits and use cases of serverless architecture. Fetching Data: Functions like getEvents and getUserRegisteredEvents for querying data from Firestore and sending it back to the client. They're efficient and cost-effective since you pay for the compute time you use. Data Modification: Functions like addKnowledgeHubDoc, removeKnowledgeHubDoc, and registerEvent show how serverless can handle CRUD operations. They allow you to add, update, or delete documents in Firestore without managing a dedicated server. Counting and Logging: The countUsers function is a good use case for serverless when need to perform operations based on database triggers, like updating a user count when a new user document is created.			
BR (E.2): Geo Location	• Map view for events using MapBox API • Non-trivial features: search places, navigate between places			
BR (E.3): Accessibility	• WCAG 2.1 AA levels • Alternative Text for Images/Non-Text Content • Keyboard Navigation • Contrast Ratio between text and background colors for users with low vision • Aria labels and roles			

Criteria	Exceeds Expectations	Meets Expectations	Needs Improvement	Fail to meet expectations
BR (E.4): Export	• Data export: csv of all events, pdf for knowledge hub documents. • Smooth and efficient user experience.			
BR (F.1): Innovation	• Bulk email: selected users. • Admin dashboard: Overview of the website including number of users and types of users. • Appointment Booking (using Calendar): The new Web Application must implement booking constraints such as booking conflict management using Calendar. • API access: 2 routes to fetch data using REST protocol			

4. Screen Recording of BRs

Create a 3-5 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).

https://drive.google.com/file/d/1eliqH5SS3kUJVxQs2yZcP6hbB5ePAw3D/view?usp=sharing

5. BR F.1 - Innovative Features

If you have implemented BR F.1, list your choices below and a brief description of how you implemented it.

0	Exact name of innovative feature (copy-paste from assignment specs). E.g. "`Bulk Email: The new Web Application must be able to send bulk email to selected users.`"	A brief, **specific** description (10-20 words) about how you implemented this in your web application. E.g. "I implemented bulk email to enable staff in my health charity website to send emails to multiple patients.	**Recommendations** for future upgrades (10-50 words)
1	Appointment Booking (using Calendar)	I implemented Appointment Booking with event register constraint using FullCakendar.io where user cannot register that clashes with another registered event.	

2	Bulk Email	I implemented bulk email using Gmail API with a table of users for selection.	
3	Admin Dashboard	I implemented an Admin Dashboard that is only accessible to admins. The dashboard provides an overview of the number and types of users in a chart.	
4	Provide API access to others	Exposed 2 routes using the REST protocol including fetching events and knowledgehub documents.	

6. Reflections: Challenges

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

As a programmer, the most challenging part of this assignment is to learn and research all different kinds of technology to meet the requirement for example finding the API for email sending. I was trying to set up with SendGrid but failed and wasted lots of time. Moreover, another challenging part is to deploy both the server and the static website. Both CloudFlare and Firebase hosts do not directly support hosting backend node.js with the front end. So moving from the backend locally to the cloud is challenging. However, I learnt a lot from these experiences via making mistakes and the documentation of firebase/cloudflare. I gained understanding on the whole development process from front, back and cloud.

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
Example: ChatGPT for brainstorming ideas	*I used ChatGPT to brainstorm how to do X because I was feeling stuck with Y problem.*

ChatGPT for debugging	I used ChatGPT to debug and explain the error messages when it is complicated and hard to understand.
Microsoft copilot	I used copilot to get ideas and understandability of documentation of new technologies such as FullCalendar.io and Firebase Cloud function.
ChatGPT for mock data generation	I used ChatGPT to generate fake data for events and knowledehub documents.