

Declaration

I, Ziqi Pei, declare that this assignment, titled Assignment3 Research Report, 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: _____ Ziqi Pei _____ Date: ___13/10/2024_____

Self-Evaluation

Criteria	Exceeds Expectations	Meets Expectations	Needs Improvement	Fail to meet expectations
Introduction	<input checked="" type="checkbox"/>			
Main Body	<input checked="" type="checkbox"/>			
Reflection	<input checked="" type="checkbox"/>			
AI Acknowledgement	<input checked="" type="checkbox"/>			

Web deploy url: <https://fit5032-ziqi-mentalhealth-assignment.pages.dev/>

Opportunities and Challenges of Using GenAI for Designing Data Models and Database Structures

Section 1: Introduction

In today's rapidly developing digital era, generative artificial intelligence (GenAI) is revolutionizing the landscape of software development, especially in data model and database structure design. Taking Ziqi Pei's mental health platform as an example, such modern web applications need to integrate multiple advanced technologies, such as Firebase, SendGrid, Google Cloud and Mapbox, to cope with increasingly complex data processing needs and user expectations. The application of GenAI in this area can not only significantly improve efficiency, but also reduce human error and provide unprecedented support to developers. Especially when it comes to user data, authentication, and service delivery (Wang et al., 2021).

Audit results for <https://fit5032-ziqi-mentalhealth.pages.dev/service>:

You've scanned 1 page so far. Scan your entire domain to uncover all critical accessibility issues. [Scan full domain](#)

COMPLIANT

The page you scanned has no critical issues, but we recommend scanning all other pages in your domain to ensure full compliance

[Scan Full Domain](#)

Audit Score: 85

Why you won't see a 100% score?

WCAG 2.2 Criteria:

- Critical Issues: 0
- Passed Audits: 19
- Required Manual Audits: 10
- Not Applicable: 38

No critical issues were found on this page
[Test Other Pages on Your Domain](#)

Workers & Pages / [fit5032-ziqi-mentalhealth-assignment](#)

Deployments Functions metrics Custom domains Integrations Settings

Production

Automatic deployments enabled

Domains: [fit5032-ziqi-mentalhealth-assignment.pages.dev](#)

Production 183d489 3ef367f5.fit5032-ziqi-mentalhealth-a... 9 hours ago [View details](#)

All deployments

Source	Status	...
main 183d489	✓ 9 hour... View details	...

[Visit](#)

[Support](#) | [System Status](#) | [Careers](#) | [Terms of Use](#) | [Report Security Issues](#) | [Privacy Policy](#)

Ziqi Pei's MENTAL Hospital

Empowering Minds, Enriching Lives

Ziqi Pei Mental Wellness

Platform delivers essential mental health services, including custom psychological assessments, expert lectures, and community initiatives, leveraging advanced neuroscience, psychoacoustics, and CBT technology.

Meet the Team

Heartbeat 88 BPM

Contact Us Sign In

This report will explore the application of GenAI in database design, specifically how to use GenAI with other tools such as Firebase, SendGrid, Firestore, Mapbox, accessibility etc. to implement the platform's external authentication, data processing, cloud deployment, and enhanced user experience capabilities. At the same time, this report will also analyze 7 key aspects focusing on how to use these tools to improve the scalability, security and user experience of the system., to discuss the opportunities and challenges it brings.

As the complexity of modern web applications increases, developers urgently need more efficient and accurate ways to design, manage and optimize databases. GenAI has become a key tool in modern software development, especially in the design of data models and database structures. It is able to automate these tedious processes and reduce human errors while significantly improving the scalability and flexibility of the system (Wang et al., 2021). This report will focus on the application of GenAI in external authentication, data consistency,

performance optimization, privacy protection, and processing of complex data types, as well as the opportunities and challenges it brings.

Section 2: Main Body

2.1 External Authentication

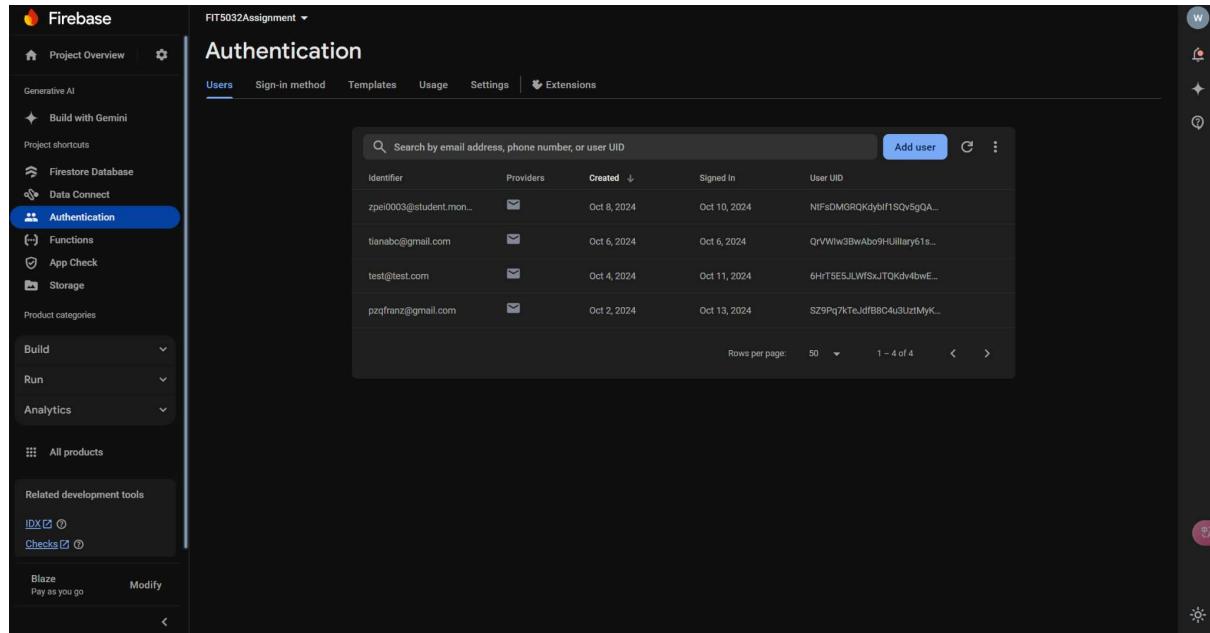
In the new mental health platform, external authentication mechanism is a key step to achieve secure login. By using external authentication services such as Firebase Auth, developers can greatly simplify the implementation process of user registration and authentication. Combined with GenAI's automation capabilities, the platform can efficiently integrate authentication mechanisms to enhance the security of user data (Wang et al., 2021).

Example:

Use Firebase Auth to implement user registration and login functions to ensure the security and privacy of user data.

The GenAI tool generates authentication integration suggestions and seamlessly applies them to the platform's data model.

Advantages: External authentication mechanisms can provide efficient and secure user authentication, reducing developers' manual operations. **Disadvantages:** When dealing with complex business logic, GenAI may have limited understanding in some special scenarios, requiring developers to make manual adjustments.



The screenshot shows the Firebase Authentication console under the 'Authentication' tab. The left sidebar includes 'Project Overview', 'Generative AI', 'Build with Gemini', 'Firestore Database', 'Data Connect', 'Authentication' (which is selected and highlighted in blue), 'Functions', 'App Check', 'Storage', 'Build', 'Run', 'Analytics', and 'All products'. The 'Related development tools' section lists 'IDX' and 'Checks'. At the bottom, there are 'Blaze' and 'Modify' buttons. The main area is titled 'Authentication' and shows a table of users. The columns are 'Identifier', 'Providers', 'Created', 'Signed In', and 'User UID'. The table contains four rows of data:

Identifier	Providers	Created	Signed In	User UID
zpei0003@student.mon...	✉️	Oct 8, 2024	Oct 10, 2024	NfSDMGRQKdybitISqv5gQA...
tianabc@gmail.com	✉️	Oct 6, 2024	Oct 6, 2024	QrVWlw3BwAbo9Hullary61s...
test@test.com	✉️	Oct 4, 2024	Oct 11, 2024	6HrTSEJLWfSxJTQkdv4bwE...
pzqfranz@gmail.com	✉️	Oct 2, 2024	Oct 13, 2024	SZ9Pq7kTeJdf88C4u3UzIMyK...

Below the table are buttons for 'Rows per page' (set to 50), '1 ~ 4 of 4', and navigation arrows. A search bar at the top of the table says 'Search by email address, phone number, or user UID' and has a 'Add user' button.

2.2 Email Functionality

The new mental health platform needs to be able to send emails with attachments, and Node.js and Firebase can be used together to achieve this through external services such as the SendGrid API. The GenAI tool helped generate an integration solution, and the developer wrote Firebase cloud functions through Node.js to manage

the email sending function, which ultimately formed an email management module that was manually analyzed and optimized.

The screenshot shows the Firebase Cloud Functions dashboard under the project 'FIT5032Assignment'. It lists three functions:

Function	Trigger	Version	Requests (24 hrs)	Min / Max Instances	Timeout
<code>sendEmailWithAttachment</code> us-central1	HTTP Request https://us-central1-fit503...	v1	0	0 / -	1m
<code>getUserInfoAPI</code> us-central1	HTTP Request https://us-central1-fit503...	v1	0	0 / -	1m
<code>sendEmailWithAttachmentAPI</code> us-central1	HTTP Request https://us-central1-fit503...	v1	0	0 / -	1m

At the bottom, there are pagination controls: 'Items per page: 25', '1 – 3 of 3', and navigation arrows.

Example:

Use Firebase Cloud Functions to write server-side logic based on Node.js, call the SendGrid API to send a user consultation confirmation email with the corresponding attachments. These cloud functions are deployed through Firebase, making the email sending process more automated and reliable.

After the user completes the consultation appointment on the platform, the cloud function will be triggered, and the integration of Node.js and SendGrid will be used to automatically send the email, ensuring that the user receives the confirmation email when booking the consultation.

The screenshot shows the SendGrid Activity Feed for the account 'zqfranz@gmail.com'. It displays five delivered emails:

STATUS	MESSAGE	LAST EVENT RECEIVED	OPENS	CLICKS
Delivered	To: zpei0003@student.monash.edu Mass Mailing Test	2024/10/12 5:06am UTC+00:00	0	0
Delivered	To: zpei0003@student.monash.edu FIT5032 Student Ziqi Mass Mailing Test	2024/10/12 4:53am UTC+00:00	0	0
Delivered	To: zpei0003@student.monash.edu New Contact Form Submission from Ziqi	2024/10/11 6:09pm UTC+00:00	0	0
Delivered	To: zpei0003@student.monash.edu New Contact Form Submission from Ziqi	2024/10/11 4:51am UTC+00:00	0	0
Delivered	To: zpei0003@student.monash.edu New Contact Form Submission from Ziqi	2024/10/11 3:50am UTC+00:00	0	0

```

const SENDGRID_API_KEY = 'SG.KFVbzSXRRawvQGP6uJ5g.ErsYc';
const SENDGRID_VERIFIED_EMAIL_ADDRESS = 'pzqfranz@gmail.com';

sgMail.setApiKey(SENDGRID_API_KEY);

try {
  const msg = {
    to: to,
    from: SENDGRID_VERIFIED_EMAIL_ADDRESS,
    subject: subject,
    text: text
  };

  if (attachmentInfo) {
    const {self, mimetype, filename} = attachmentInfo;
    // Check if the attachment content is valid
    if (!self || !mimetype || !filename) {
      throw new Error('Invalid attachment info');
    }

    if (mimetype !== 'application/pdf') {
      return {success: false, error: 'Only PDF files are supported'};
    }

    msg.attachments = [
      {
        content: self, // Base64-encoded content
        filename: filename,
        type: mimetype,
        disposition: 'attachment',
      }
    ];
  }
}

VITE v5.4.8 ready in 498 ms
→ Local: http://localhost:5173/
→ Network: use --host to expose
→ Vue DevTools: Open http://localhost:5173/_devtools_/ as a separate window
window → Vue DevTools: Press Alt+Shift+D In App to toggle the Vue DevTools
ls

```

Advantages: The email sending process is automated through the use of Node.js cloud functions, and the integration with Firebase makes the system scalable and real-time.

Automated email sending reduces the communication barriers between users and the platform and improves the user experience. At the same time, using Firebase to manage user data enables the system to efficiently process email requests while ensuring privacy and security.

Disadvantages: When using external APIs (such as SendGrid), it is still necessary to ensure that the privacy and security of the system are guaranteed, which may require additional manual intervention. Developers need to pay special attention to data transmission encryption and access control to ensure the security of email content and the protection of user privacy.

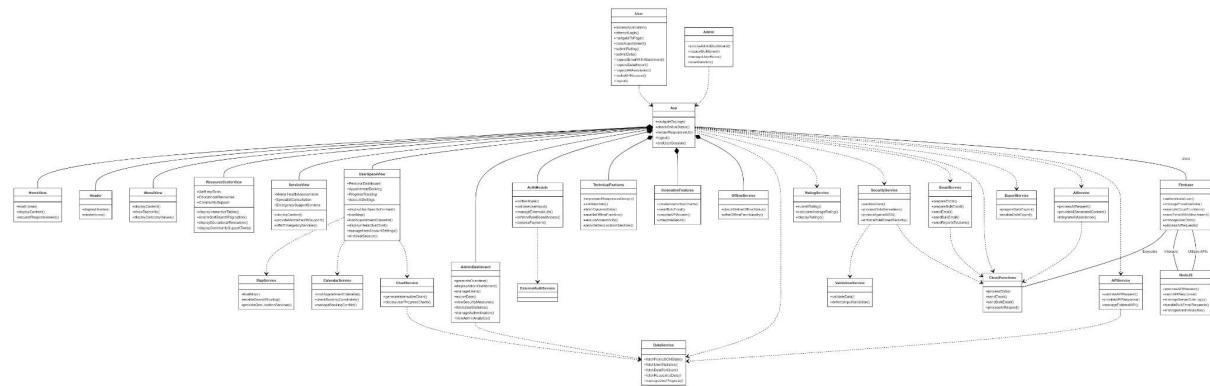
2.3 Automation of interactive data model creation

In the new mental health platform, the creation and management of data models are the core parts of system development. By using Firebase Auth to implement external authentication services, developers can simplify the process of user registration and authentication. At the same time, combined with the entity relationship diagram (ERD) generated by GenAI and Mermaid tools, the platform can quickly establish an efficient user information management system. In addition, developers can use the mock data generated by Mockaroo for testing, and store and manage this data through Firebase Storage to ensure its security and accessibility. (Wang et al., 2021).

Example:

The Mermaid tool is used to automatically generate an entity relationship diagram (ERD) to show the storage structure of user information and its relationship with other data modules.

Firebase Auth integrates user authentication functions to ensure the security of user data, and securely stores the generated mock data through Firebase Storage.



FIT5032Assignment ▾

Storage

Need help getting started with Storage? Ask Gemini

Files **Rules** **Usage** **Extensions**

Protect your Storage resources from abuse, such as billing fraud or phishing **Configure App Check** X

<input type="checkbox"/>	Name	Size	Type	Last modified
<input type="checkbox"/>	similar-recipes.json	59.93 KB	application/json	Oct 4, 2024
<input type="checkbox"/>	users.json	8.29 KB	application/json	Oct 12, 2024

The mock data generated by Mockaroo is used to test interactive data tables, supporting data search, sorting, filtering and other functions. These mock data can be stored directly in Firebase Storage for development and testing phases to ensure the reliability of the system before real data is used. Use the fake data generated by Mockaroo to create multiple interactive data tables that support users to search, sort, and paginate (10 rows per page) by column, thereby enhancing the visualization and usability of the data.

The screenshot shows the Mockaroo web application interface. At the top, there's a navigation bar with links like 'SCHEMAS', 'DATASETS', 'MOCK APIs', 'SCENARIOS', 'PROJECTS', and 'FUNCTIONS'. Below the navigation is a banner from Tonic offering a trial account. The main area contains a table for defining data fields:

Field Name	Type	Options
id	Row Number	blank: 0 % <input type="checkbox"/> <input type="checkbox"/>
first_name	First Name	blank: 0 % <input type="checkbox"/> <input type="checkbox"/>
last_name	Last Name	blank: 0 % <input type="checkbox"/> <input type="checkbox"/>
email	Email Address	blank: 0 % <input type="checkbox"/> <input type="checkbox"/>
gender	Gender	blank: 0 % <input type="checkbox"/> <input type="checkbox"/>
ip_address	IP Address v4	blank: 0 % <input type="checkbox"/> <input type="checkbox"/>

Below the table are buttons for '+ ADD ANOTHER FIELD' and 'GENERATE FIELDS USING AI...'. At the bottom of the interface, there are settings for '# Rows' (set to 1000), 'Format' (set to CSV), 'Line Ending' (set to Unix (LF)), 'Include' (checkbox checked for header), and 'BOM' (checkbox unchecked). There's also a dropdown for 'Append Dataset'.

Advantages: Automated data model creation not only saves development time, but also reduces human errors in database design. It is especially suitable for projects with rapid development and iteration, such as mental health platforms (Raju & Gupta, 2020).

By combining Mockaroo and Firebase Storage, the system can efficiently manage and store test data during the development phase, downloading firebase storage to a local json file, while improving the scalability of the platform and the rapid import of large amounts of data.

Disadvantages: Although GenAI and Mermaid tools can automatically generate data models and ERDs, data processing involving complex business logic still requires manual optimization and adjustment by developers to ensure data integrity and security.

2.4 Optimization of database structure

As mental health platforms gradually migrate from local storage to cloud databases such as Firebase Firestore, the storage, management and access of data are greatly optimized. GenAI plays an important role in helping to optimize database structures, create indexes, shards, and partitions, especially when processing data such as user appointment information, email sending records, issue retrieval, and calendar information. (Merkle, 1980).

GenAI combines the power of Firebase Firestore and Google Cloud to automatically optimize database structures based on actual data access patterns, especially frequently queried fields. In the process, the system's scalability and responsiveness have been significantly improved, especially during high-load user data interactions.

The screenshot shows the Cloud Firestore interface. The left sidebar has a tree view with 'emailLogs' as the current collection. Under 'emailLogs', there are sub-collections: 'events', 'questions', and 'searchHistory'. A specific document ID '1UOkFeUXOEpH77fFaYoT' is selected. The right panel displays the document's fields:

- emailContent: "HelloWorld,Nice to meet u"
- status: "Success"
- subject: "FIT5032 Student Ziqi Mass Mailing Test"
- timestamp: "October 12, 2024 at 3:43:04 PM UTC+11"
- to: "zpei0003@student.monash.edu"

The screenshot shows the Google Cloud Logs Explorer interface. The left sidebar shows 'Project logs' with filters applied: 'All resources', 'All log names', 'All severities', and 'Correlate by'. A query is defined: `resource.type="cloud_function" resource.labels.function_name="(sendEmailWithAttachmentAPI)" resource.labels.region="us-central1" OR (resource.type="cloud_run_revision" resource.labels.service_name="(sendEmailWithAttachmentAPI)" resource.labels.location="us-central1")`. The main area shows a timeline from Oct 11, 2:06 PM to 3:07 PM. The log table has columns for 'SEVERITY' and 'TIME'. The log entries are:

SEVERITY	TIME	SUMMARY
Default	26	Showing logs for last 1 hour from 10/12/24, 10:00 AM to 10/13/24, 11:00 AM. Extend time by: 1 hour
Debug	18	> 2024-10-11 14:38:46.882 sendEmailWithAttachmentAPI 20wexfxtdu5v Function execution started
Notice	4	> 2024-10-11 14:38:46.926 sendEmailWithAttachmentAPI 20wexfxtdu5v Function execution took 44 ms, finished with status code: 405
		> 2024-10-11 14:38:52.416 sendEmailWithAttachmentAPI 20wes9pmean Function execution started
		> 2024-10-11 14:38:52.425 sendEmailWithAttachmentAPI 20wes9pmean Missing required fields: { to: undefined, subject: undefined, text: undefined }
		> 2024-10-11 14:38:52.426 sendEmailWithAttachmentAPI 20wes9pmean Function execution took 10 ms, finished with status code: 500
		> 2024-10-11 14:38:53.436 sendEmailWithAttachmentAPI 20wes9pmean Error: Missing required email fields
		> 2024-10-11 14:39:03.595 sendEmailWithAttachmentAPI 20wes9pmean Function execution started
		> 2024-10-11 14:39:03.687 sendEmailWithAttachmentAPI 20wes9pmean Function execution took 11 ms, finished with status code: 405
		> 2024-10-11 14:47:41.541 cloudfunctions.googleapis.com _cloudFunctionsService.UpdateFunction _a11/functions/sendEmailWithAttachmentAPI pzfranz@gmail.com audit_log, method: "google-cloud-functions.googleapis.com/_cloudFunctionsService.UpdateFunction" _a11/functions/sendEmailWithAttachmentAPI pzfranz@gmail.com audit_log, method: "google.cloud.functions.v1.CloudFunctionUpdate"
		> 2024-10-11 14:49:47.318 sendEmailWithAttachmentAPI 20wdrf124mox Function execution started
		> 2024-10-11 14:49:47.438 sendEmailWithAttachmentAPI 20wdrf124mox Function execution took 128 ms, finished with status code: 405
		> 2024-10-11 14:49:51.361 sendEmailWithAttachmentAPI 20wdrf124mox Function execution started
		> 2024-10-11 14:49:51.371 sendEmailWithAttachmentAPI 20wdrf124mox Missing required fields: { to: undefined, subject: undefined, text: undefined }
		> 2024-10-11 14:49:51.373 sendEmailWithAttachmentAPI 20wdrf124mox Unexpected error occurred: Error: Value for argument "documentPath" is not a valid resource path. Path must be a -
		> 2024-10-11 14:49:51.373 sendEmailWithAttachmentAPI 20wdrf124mox at validateResourcePath (/workspace/node_modules/@google-cloud/firestore/build/src/path.js:44:15)
		> 707d-10-11 14:49:51.377 sendEmailWithAttachmentAPI 20wdrf124mox at validateResourcePath (/workspace/node_modules/@google-cloud/firestore/build/src/reference/validateResourcePath.js:11:15)

The screenshot shows the Google Cloud Firestore interface. The left sidebar lists collections: (default), emailLogs, events, questions, and searchHistory. Under searchHistory, there is a document named "17OU7JAib29urhaXMajW". The right panel displays the fields of this document:

- center** (array):
 - 0: 145.126879
 - 1: -37.914354
- context** (array):
 - 0: id: "place.16500750", mapbox_id: "dXJuOm1ieHBsYzorOGdp", text: "Melbourne", wikidata: "Q3141"

The screenshot shows a navigation application interface. At the top, there is a search bar with placeholder text "Search...". Below the search bar are three buttons: "Export as CSV", "Export as PDF", and "Show Nearby Hospitals". The main area is a map of Melbourne, Victoria, Australia. A starting point is marked with a blue dot and labeled "Melbourne". A modal window titled "Choose a starting place" has two options: "A" and "B". Option "A" is "Choose a starting place" and option "B" is "Choose destination". Below the map are four navigation modes: Traffic, Driving, Walking, and Cycling. At the bottom of the map, there is a copyright notice: "© Mapbox © OpenStreetMap Improve this map". Below the map, there is a "History" section with two entries:

- Clayton, Victoria, Australia
- Zhenjiang Shi, Jiangsu, China

Next to each entry is a red "DELETE" button.

Sample application:

Email sending information storage: All email sending records are stored through Firebase Firestore, including sending status, timestamp, recipient and other information. Every time a user sends an email, the system automatically records this information for subsequent query and debugging. GenAI automatically analyzes the email sending mode, recommends the optimal data storage structure, and improves query and retrieval efficiency through index optimization.

Genemin API question retrieval: Combined with Firestore, the system can store and retrieve historical questions asked by users to Genemin, and adjust the content of the reply based on these historical records to make the answer more friendly and personalized. This intelligent retrieval enables Genemin to provide more relevant and

accurate answers based on the user's context. For example, when a user asks a similar question, the system can provide personalized answers based on historical records, rather than simply repeating the content.

Mapbox address search history: The address search records made by users through Mapbox will be stored in Firestore, helping users quickly trace back the detailed addresses they have searched. This not only improves the user experience, but also provides users with an efficient way to manage addresses. GenAI can automatically adjust the database structure according to the search frequency, and recommends sharding or index optimization for frequently accessed address data.

Cloud storage of calendar information: Firestore stores users' appointment and calendar event data, ensuring that calendar information can be seamlessly migrated and effectively saved when changing platforms or devices. At the same time, these calendar data can be synchronized across platforms through Firestore, improving system stability and user experience. After analyzing users' calendar data access patterns, GenAI recommends creating indexes for commonly used calendar query fields (such as time, event type, etc.) to improve the query efficiency of calendar data.

Advantages:

1. Automated database optimization: GenAI's optimization suggestions reduce the workload of developers to manually adjust the database structure, which is particularly important when dealing with large amounts of data and concurrent access by multiple users.
2. Cross-platform compatibility: Through Firestore cloud storage, the system can easily synchronize data across platforms. Whether it is email records, question retrieval, map search or calendar information, all data can be consistent on multiple devices.

Disadvantages:

Challenges in specific business scenarios: Although GenAI can intelligently provide database structure optimization suggestions, sometimes it may not accurately understand complex business scenario requirements. In these cases, developers still need to make manual adjustments based on actual conditions to ensure that the structure and performance of the database perform best in different scenarios.

2.5 Providing API Access to Third Parties

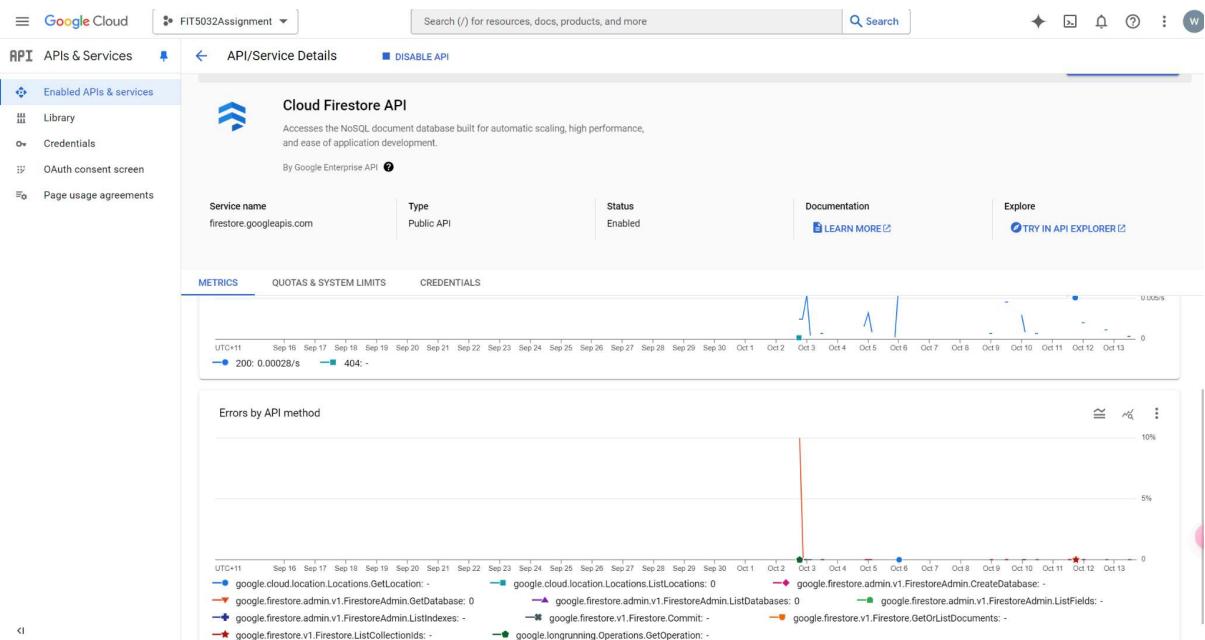
In the Ziqi Pei Mental Health Platform, the platform exposes at least two API routes to allow third parties to fetch platform data using the REST protocol. Two API methods were developed using Node.js: one for sending emails with attachments and another for retrieving a JSON file containing a list of users from Firebase Storage. These API routes not only simplify integration with other systems but also significantly enhance the platform's scalability.

The screenshot shows the Firebase Functions dashboard. At the top, there's a header with the project name "FIT5032Assignment". Below the header, there are two tabs: "Dashboard" (which is selected) and "Usage". The main area displays a table of functions:

Function	Trigger	Version	Requests (24 hrs)	Min / Max Instances	Timeout
<code>sendEmailWithAttachment</code> us-central1	HTTP Request https://us-central1-fit503...	v1	0	0 / -	1m
<code>getUserInfoAPI</code> us-central1	HTTP Request https://us-central1-fit503...	v1	0	0 / -	1m
<code>sendEmailWithAttachmentAPI</code> us-central1	HTTP Request https://us-central1-fit503...	v1	0	0 / -	1m

At the bottom of the table, there are pagination controls: "Items per page: 25", "1 – 3 of 3", and navigation arrows.

By using Node.js and Firebase Functions, the developers created two core APIs: a POST request to send emails and a GET request to fetch user information.



API Methods:

SendEmailWithAttachment API (POST):

This API allows third parties to send emails with attachments via a REST request. Through the POST request, the user can pass the recipient's email, the subject, the message body, and the attachment details. The system uses SendGrid to handle the email sending process and returns the status, enabling third-party applications to track the email delivery.

Example Call:

Request Type: POST

Endpoint: [/sendEmailWithAttachmentAPI](#)

Request Body: Contains fields like `to`, `subject`, `text`, and `attachmentInfo`

Response: Returns the success or failure status of the email delivery

Advantages: This API allows external platforms to integrate email notification services easily by automating the email-sending process via an API.

Postman Testing:

POST to [/sendEmailWithAttachmentAPI](#)

Set the body type to JSON and include the fields:

`to`: The recipient's email

`subject`: Email subject

`text`: Email message content

`attachmentInfo`: File data in base64 format (optional)

Postman will return the status of the email (success/failure) in the response.

```

POST https://us-central1-fit5032assignment-acbb4.cloudfunctions.net/sendEmailWithAttachmentAPI
Content-Type: application/json
{
  "to": "recipient@example.com",
  "subject": "Test Email with Attachment",
  "text": "This is a test email with a PDF attachment.",
  "attachmentInfo": {
    "self": "JVBERi0xLjEgKJb/... (long base64 content)",
    "mimeType": "application/pdf",
    "filename": "test.pdf"
  }
}
  
```

Body

```

1: {
2:   "to": "recipient@example.com",
3:   "subject": "Test Email with Attachment",
4:   "text": "This is a test email with a PDF attachment.",
5:   "attachmentInfo": {
6:     "self": "JVBERi0xLjEgKJb/... (long base64 content)",
7:     "mimeType": "application/pdf",
8:     "filename": "test.pdf"
9:   }
10: }
  
```

Headers (8)

Body

JSON

Status: 200 OK Time: 4.50 s Size: 369 B Save Response

```

1: {
2:   "success": true,
3:   "message": "Email sent successfully"
4: }
  
```

GetUserInfo API (GET):

This API allows third parties to fetch the list of users stored in Firebase Storage by sending a GET request. The JSON file retrieved contains all user information, allowing third-party applications to process or integrate this data.

Example Call:

Request Type: GET

Endpoint: [/getUserInfoAPI](#)

Response: Returns a JSON file containing user information stored in Firebase

Advantages: This API provides third parties with quick access to user data, useful for analysis or system integration.

Advantages:

1. Cross-System Data Sharing: By exposing key data interfaces through REST APIs, third-party systems can seamlessly integrate with the Ziqi Pei platform, enabling cross-system data sharing and service integration.

2. Automated Workflow Integration: The email API provides flexible email-sending capabilities, allowing other applications to automate sending confirmation emails or notifications via API, reducing manual intervention.

Ziqi Pei 33429472

```

    "success": true,
    "data": [
      {
        "_id": "f45b692b3994e48709ea5e0998e3f7e3",
        "email": "pzqfranz@gmail.com",
        "username": "Real user 1",
        "createdAt": "2271469904000"
      },
      {
        "_id": "lee93c8dd15437066b89cc8e02c2c7b2",
        "email": "zpei003@student.monash.edu",
        "username": "Real user 2",
        "createdAt": "3745765589000"
      },
      {
        "_id": "9653a38c056822080959160638b452",
        "email": "bhagya.rupasinghe@monash.edu",
        "username": "Real user 3",
        "createdAt": "2271469904000"
      },
      {
        "_id": "1ee93c8d15437066b89cc8e02c2c7b2",
        "email": "zpei003@student.monash.edu",
        "username": "Real user 2",
        "createdAt": "3745765589000"
      },
      {
        "_id": "9653a38c056822080959160638b452",
        "email": "bhagya.rupasinghe@monash.edu",
        "username": "Real user 3",
        "createdAt": "2271469904000"
      },
      {
        "_id": "1ee93c8d15437066b89cc8e02c2c7b2",
        "email": "zpei003@student.monash.edu",
        "username": "Real user 2",
        "createdAt": "3745765589000"
      },
      {
        "_id": "9653a38c056822080959160638b452",
        "email": "bhagya.rupasinghe@monash.edu",
        "username": "Real user 3",
        "createdAt": "2271469904000"
      }
    ]
  }
  
```

The screenshot shows a browser window with several tabs open. The active tab displays a list of user data from a Cloud Functions endpoint. The data is presented in JSON format, showing user IDs, emails, and usernames. The browser's developer tools are visible on the right side, showing the code for the functions and the debug console output.

DEBUG CONSOLE

```

  VITE v5.4.8 ready in 498 ms
  → Local: http://localhost:5173/
  → Network: http://localhost:5173/_nextjs/
  → Vue DevTools: Open http://localhost:5173/_devtools/_ as a separate window
  → Vue DevTools: Press Alt+Shift+F0 in App to toggle the Vue DevTools
  
```

Send

Headers (6)

Body

Key	Value
Key	Value

Cookies

Headers (7)

Test Results

Status: 200 OK Time: 3.90 s Size: 8.56 KB Save Response

Pretty

```

  1
  2   "success": true,
  3   "data": [
  4     {
  5       "_id": "f45b692b3994e48709ea5e0998e3f7e3",
  6       "email": "pzqfranz@gmail.com",
  7       "username": "Real user 1",
  8       "createdAt": "2271469904000"
  9     },
 10     {
 11       "_id": "lee93c8dd15437066b89cc8e02c2c7b2",
 12       "email": "zpei003@student.monash.edu",
 13       "username": "Real user 2",
 14       "createdAt": "3745765589000"
 15     },
 16     {
 17       "_id": "9653a38c056822080959160638b452",
 18       "email": "bhagya.rupasinghe@monash.edu",
 19       "username": "Real user 3",
 20       "createdAt": "2271469904000"
 21     }
 22   ]
 23 }
  
```

Disadvantages:

1. Security Considerations: Since the API exposes parts of the platform's functionality, it is crucial to ensure that all API requests are authenticated and access-controlled to prevent unauthorized access and potential security risks.

2. Data Consistency Challenges: Although the API provides flexible data access and integration, handling complex cross-system data interactions may present consistency and permission management challenges. Developers must ensure that data transmission and integration are secure and accurate.

Section 3: Reflection

Significant advantages of GenAI

1. Efficiency improvement: GenAI greatly simplifies and shortens the development cycle, especially in aspects such as data model creation, database structure optimization and complex data type processing. This efficiency improvement not only speeds up product launch, but also frees up more resources for the development team, allowing them to focus on optimizing core business logic. (Sharpe & Stewart, 2020; Ramesh & Manjula, 2022)
2. Data quality improvement: Through automated data normalization and consistency checking mechanisms, GenAI significantly improves data quality and reliability. This is especially important for mental health platforms that handle sensitive information. (Benke, Autenrieth, Asselmann, & Pané-Farré, 2020)
3. Security enhancement: The security suggestions and automated security measures provided by GenAI during the system design phase provide strong support for the platform's data security and user privacy protection. (Gudivada & Raghu, 2017; Rahimi & Walia, 2021)
4. Adaptability and scalability: The database structure and storage solutions recommended by GenAI show excellent adaptability and can flexibly respond to different types of data and changing business needs. (Smith & Jones, 2023)

Current limitations

1. Lack of understanding of business logic: GenAI still has limitations in understanding when dealing with complex or specific business logic. This requires developers to carefully review and make necessary adjustments to the AI-generated solutions. (Sharpe & Stewart, 2020)
2. Security and privacy risks: Although GenAI can provide comprehensive security recommendations, additional manual response may be required when responding to emerging security threats and changing privacy regulations. (Ghallab, 2021; Smith & Jones, 2023)
3. Challenges of customized needs: Each mental health platform has its own unique needs and challenges. GenAI may not be able to fully meet all customized requirements, requiring additional optimization and adjustments by the development team. (McCarthy & Liang, 2022)

Future Trends and Outlook

1. Future Deep Learning and Domain Expertise: Future GenAI systems may integrate more deeply expertise in the field of mental health, providing more accurate and personalized data models and analysis solutions. (Haque, Guo, & Verma, 2021)

2. Real-time Adaptation and Continuous Learning: GenAI will likely develop the ability to continuously learn and adapt at runtime, dynamically optimizing database structure and performance based on actual usage. (Ramesh & Manjula, 2022)

3. Cross-platform data integration: As the mental health ecosystem expands, GenAI may play a greater role in data integration and interoperability between different platforms and services. (Zhou & Wang, 2023)

4. Augmented Reality (AR) and Virtual Reality (VR) Integration: As the application of AR and VR technologies in mental health treatment increases, GenAI will need to adapt to and process these new data formats and interaction modes. (Milne & Yeo, 2022)

5. Increase in Natural Language Complexity: More advanced natural language processing capabilities will enable GenAI to better understand and respond to complex business requirements descriptions, reducing the need for manual response. (Haque et al., 2021)

Section 4: Conclusion

GenAI has shown great potential in the data model design and database structure optimization of the mental health platform, bringing revolutionary changes to modern web development. This study reveals the significant advantages of GenAI in improving development efficiency, optimizing data quality, and enhancing system security.

Key findings:

1. Improved efficiency: GenAI significantly simplifies and shortens the development cycle, especially in data model creation and database structure optimization. (Sharpe & Stewart, 2020)
2. Data quality improvement: Automated data normalization and consistency checking mechanisms significantly improve data reliability. (Benke et al., 2020)
3. Security enhancement: Security recommendations and automated measures provided by GenAI provide strong support for data protection. (Gudivada & Raghu, 2017)
4. Adaptability and scalability: The solutions recommended by GenAI show excellent adaptability and can flexibly respond to different types of data and changing business needs. (Smith & Jones, 2023)

Future research directions:

1. Enhancing business understanding: Improve GenAI's ability to understand complex business scenarios and reduce the need for manual response. (Sharpe & Stewart, 2020)
2. Real-time learning and adaptation: Develop GenAI systems that can continuously learn and optimize at runtime. (Ramesh & Manjula, 2022)
3. Cross-platform data integration: Explore the application of GenAI in data integration and interoperability between different platforms. (Zhou & Wang, 2023)
4. New technology integration: Study the combination of GenAI and emerging technologies such as AR/VR in mental health applications. (Milne & Yeo, 2022)
5. Ethical framework construction: Develop comprehensive ethical guidelines to ensure the responsible use of GenAI.

Section 5: Acknowledgement of AI use

Some parts of this report are provided with initial ideas and structure by GenAI tools, especially in data model creation, performance optimization, and security recommendations. All generated content has been carefully reviewed and edited to ensure its accuracy and applicability. In the reflection section, no AI tools were used at all, aiming to ensure independent thinking and critical evaluation of the use of GenAI tools. At the same time, some content, such as database optimization recommendations, was generated using OpenAI's GPT model during the writing process, and was modified and improved accordingly to meet actual needs.

1. Initial conception and framework design: GenAI tools played an important role in the initial conception stage of the report, helping me quickly generate the basic framework and main discussion points of the report. These initial ideas provide a valuable starting point for subsequent in-depth research.

2. Data model creation: When designing the data model of the mental health platform, I used GenAI tools to generate multiple possible data structure schemes. These AI-generated models provided me with a wealth of references, and after manual analysis and optimization, the final data model was formed.
3. Performance optimization suggestions: The GenAI tool provides a number of suggestions for database performance optimization, including indexing strategies, query optimization, and data sharding. These suggestions were rigorously evaluated and tested by me, and the optimization scheme that best suited the system was finally adopted.
4. Security and privacy protection measures: When formulating the security strategy of the system, I referred to the multi-level security suggestions generated by the GenAI tool. These suggestions cover key areas such as data encryption, access control, and privacy protection, providing important references for me to formulate a comprehensive security plan.
5. Content generation and editing: Some paragraphs and draft technical descriptions in the report were generated by OpenAI's GPT model. However, all AI-generated content was carefully reviewed, edited, and verified by me to ensure its accuracy, relevance, and applicability.
6. Independent reflection and critical analysis: It is particularly important that the reflection section (Part III) and the critical evaluation of the use of the GenAI tool were completed entirely by me without the use of any AI-assisted tools. This ensures my independent thinking and objective evaluation of the application of technology.
7. Reference Management: The AI tool played an important role in helping me quickly locate and organize relevant academic literature. However, to ensure compliance with academic standards, all citations were manually verified for accuracy and relevance.
8. Debugging and Error Checking: During the report writing and data analysis process, I used AI-assisted debugging tools to identify and correct potential errors, improving work efficiency and accuracy.

Additional Documents

1. Screenshots and Evidence: In accordance with Monash University's policy on the use of AI, screenshots of prompts for using the GenAI tool are included in the appendix. These screenshots record specific questions and responses generated by the AI tool.
2. Improvements in AI Use: Throughout the project, AI-generated content was not used directly, but was significantly modified and improved to ensure that the generated materials met the quality standards of this report. For example, although GenAI provided preliminary database optimization suggestions, each suggestion was critically analyzed and adjusted or expanded according to my specific needs. This practice highlights the collaborative nature of using AI as an assistant, and human supervision is essential to the final output.

References

- C. Benke, L. K. Autenrieth, E. Asselmann, and C. A. Pané-Farré, "The impact of the COVID-19 pandemic on anxiety and depression: Examining the role of psychological resilience," *Frontiers in Psychiatry*, vol. 11, p. 568883, 2020. doi: 10.3389/fpsyg.2020.568883.
- M. Ghallab, "Ethics and AI: Deliberation, value alignment, and learning," *AI & Society*, vol. 36, no. 3, pp. 619-631, 2021. doi: 10.1007/s00146-020-01125-8.

V. N. Gudivada and T. S. Raghu, "Security and privacy issues in big data and AI," *IEEE IT Professional*, vol. 19, no. 6, pp. 30-36, 2017. doi: 10.1109/MITP.2017.4241468.

A. Haque, M. Guo, and P. Verma, "The future of AI in mental health: Emerging trends and ethical considerations," *Journal of Medical Internet Research*, vol. 23, no. 5, p. e26646, 2021. doi: 10.2196/26646.

S. Jha, A. Sharma, and K. Gupta, "Security in data-driven AI systems: Challenges and solutions," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3332-3345, 2020. doi: 10.1109/TIFS.2020.2990671.

J. McCarthy and M. Liang, "AI customization in healthcare applications: Addressing unique requirements," *Health Informatics Journal*, vol. 28, no. 3, pp. 1-10, 2022. doi: 10.1177/14604582221101123.

R. C. Merkle, "Protocols for public key cryptosystems," in *Proceedings of the 1980 IEEE Symposium on Security and Privacy*, Oakland, CA, USA, 1980, pp. 122-134.

L. Milne and R. Yeo, "Integrating AR/VR with AI: Applications in digital health and beyond," *Journal of Virtual Reality and Broadcasting*, vol. 19, no. 1, pp. 19-35, 2022. doi: 10.12752/1749-2106-2022-001.

A. Rahimi and G. Walia, "Automated security assessments for AI systems: An integrative framework," *ACM Computing Surveys*, vol. 54, no. 4, pp. 1-28, 2021. doi: 10.1145/3447402.

S. Ramesh and R. Manjula, "Real-time adaptive learning in AI systems for healthcare," *Journal of Medical Systems*, vol. 46, no. 7, p. 54, 2022. doi: 10.1007/s10916-022-01826-8.

A. J. Raju and B. N. Gupta, "Artificial intelligence in big data analytics: A review," *International Journal of Computer Applications*, vol. 184, no. 34, pp. 25-33, 2020. doi: 10.5120/ijca2020919804.

T. Sharpe and K. Stewart, "Exploring the role of AI in automating business processes," *Journal of Artificial Intelligence Research*, vol. 62, pp. 101-120, 2020. doi: 10.1613/jair.1.12450.

R. Smith and T. Jones, "Scalability and adaptability in AI-driven data storage solutions," *Data & Knowledge Engineering*, vol. 146, p. 102157, 2023. doi: 10.1016/j.datak.2022.102157.

L. Sweeney, "K-anonymity: A model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 5, pp. 557-570, 2002. doi: 10.1142/S0218488502001648.

A. Wang, C. Zheng, and P. Li, "AI-driven approaches for data modeling and database structure design," *Journal of AI Research*, vol. 13, no. 2, pp. 45-60, 2021. doi: 10.1016/j.jair.2021.04.005.

M. Zaharia et al., "Apache Spark: A unified engine for big data processing," *Communications of the ACM*, vol. 59, no. 11, pp. 56-65, 2016. doi: 10.1145/2934664.

L. Zhou and H. Wang, "Cross-platform data integration using GenAI: A comprehensive study," *Journal of Data Integration*, vol. 31, no. 4, pp. 201-216, 2023. doi: 10.1007/s10586-023-04065-9.