

**REPUBLIC OF
CAMEROON**

**MINISTRY OF HIGHER
EDUCATION**

Peace-Work-Fatherland



REPUBLIC DU CAMEROUN

**MINISTERE DE
L'ENSEIGNEMENT
SUPERIEUR**

Paix-Travail-Patrie

UNIVERSITY OF BUEA

FACULTY OF ENGINEERING AND TECHNOLOGY,

CEF440:

INTERNET PROGRAMMING AND MOBILE PROGAMING

GROUP 23: TASK 6

Names	Matriculation Number
NDIFON LEMUEL ASHU-MBI	FE21A247
NGWASIRI RYAN TANIFIORM ONGA	FE21A266
NJIDDA SALIFU	FE21A272
ASONGNKWELLE COURAGE AYIM	FE21A142
NKWO BRAINIE NGONDA	FE21A282

Course Instructor:

June 2024,

Dr. Nkemeni Valery

Table of Content

1. Introduction	1
1.1 Overview	1
1.2 Importance of Database Design	1
1.3 Key Considerations	2
2. Database Technology Stack	2
3. Entity Relationship	3
i. Admin Entity	3
ii. Session Entity	4
iii. Student Entity	4
iv. Attendance Entity	5
Entity Relations	5
Figure 1 : Entity Relationship Diagram	6
4. Relational Schema	7
5. Firebase Implementation	8
5.1 Admin Collection	8
5.2 Student Collection	9
5.3 Session Collection	10
5.4 Attendances Collection	11
6. Conclusion	11
7. References	12

1. Introduction

1.1 Overview

The Attendance Report Generator app is designed to streamline the process of tracking and generating attendance reports for educational courses. This mobile application allows users to input specific details such as course code, session time, and date to generate detailed attendance reports. These reports can then be exported in PDF format and stored on the user's device for easy access and sharing.

1.2 Importance of Database Design

Database design is a crucial aspect of any application that manages data. In the context of the Attendance Report Generator app, an efficient and well-structured database is essential for several reasons:

- ✓ **Data Integrity and Accuracy:** Proper database design ensures that the attendance records are stored accurately, without duplication or errors. This is vital for generating reliable reports.
- ✓ **Performance Optimization:** Efficient querying and retrieval of data are necessary for providing a seamless user experience. Well-designed databases enable quick access to relevant records, even as the dataset grows.
- ✓ **Scalability:** As the app usage increases, the database should be able to handle more data and more complex queries without performance degradation.
- ✓ **Maintainability:** A clear and logical database structure makes it easier to maintain and update the app, allowing for future enhancements and bug fixes.

1.3 Key Considerations

In designing the database for the Attendance Report Generator app, several key considerations were taken into account:

- ✓ **Normalization:** Ensuring that the database is normalized to avoid redundancy and ensure data integrity.
- ✓ **Relationships:** Clearly defining relationships between different types of data (e.g., courses and attendance records) to allow for efficient data retrieval.
- ✓ **Indexing:** Implementing appropriate indexing strategies to speed up query performance for common operations such as fetching attendance records based on course codes and dates.
- ✓ **Security:** Protecting sensitive student data by implementing access controls and secure data transmission protocols.

2. Database Technology Stack

The app uses Firebase Firestore as its backend database service.

Firebase Firestore is a flexible, scalable, and real-time NoSQL database that is well-suited for mobile applications. It offers several advantages:

- ✓ **Real-time Synchronization:** Changes to the data are instantly reflected in the app, providing real-time updates.
- ✓ **Scalability:** Firestore can handle large volumes of data and high query loads, making it suitable for growing applications.
- ✓ **Ease of Use:** Firestore integrates seamlessly with other Firebase services and provides an easy-to-use API for data operations.
- ✓ **Security Rules:** Firestore allows for fine-grained access control through security rules, ensuring that only authorized users can access or modify data.

3. Entity Relationship

Entity relationship is a design approach that establishes the relationship between main entities and gives description on their various attributes for each entity.

Entity relationships brings clarity and understanding in relationships among entities easing the design process in context of data handling and data organizing in the database design of the concerned system.

In the context of our fingerprint student attendance app, there are various relationships that exist between various entities. According to the analysis performed at the requirement analysis phase, there are five main entities that exist within this subject bringing about all entity relationships and their attributes which include:-

- ✓ Admin Entity
- ✓ Session Entity
- ✓ Student Entity
- ✓ Attendance Entity

The following points clearly describe each entity and lists all the attributes in concern

i. Admin Entity

The admin entity is responsible for registering student data and their biometric data. This entity is also responsible for adding courses students are offering along with their periods and time slots. This entity is mainly the input gateway to our application as admin input are the basis for the application's functionality.

The admin entity has various attributes which are:-

- ✓ admin_id (auto-generated data in the database for extra security)
- ✓ Username (key attribute)
- ✓ password

ii. Session Entity

The session entity is an object entity created by the admin. This entity holds information about the courses offered by the students in the attendance taking process. There are attributes that are possessed by this entity which are

- ✓ course_id
- ✓ course_code
- ✓ course_name
- ✓ date
- ✓ time

iii. Student Entity

The student entity represents the primary users of the system. They will be solely responsible to mark attendances by authenticating their biometric fingerprint scanned taken by the admin on selecting the course, the course day and course time.

The action of taking their attendances in turn generates a new entity called the attendance entity which the report can be generated by the admin entity.

Upon successful student registration, the system generates a unique identifier, this is a four character alphabetic string unique to every student. This attribute is due to the fact that according to data protection laws, fingerprint data can't be accessed or stored in a database system, it can only be stored in the mobile device's hardware.

Due to this fact the database doesn't contain the fingerprint data. Instead, the fingerprint's data capture can still be tracked in the database. It can be tracked using a fingerprintData entity that stores a boolean value (true or false) indication if fingerprint enrollment is successful or not.

The attributes of this entity include:-

- ✓ std_id (auto-generated by the database)
- ✓ std_name
- ✓ matricule (key attribute)
- ✓ fingerprintData
- ✓ unique_identifier

iv. Attendance Entity

This entity is created as a result of students taking attendance. This entity will comprise data from the student entity(knowing which student takes attendance) , the course entity (for which course it was for) and the periods entity reports for particular courses by the admin.

The attributes of this entity include:-

- ✓ attend_id (auto-generated by the database for security)
- ✓ std_name
- ✓ matricule
- ✓ course_day
- ✓ course_time
- ✓ course_name

Entity Relations

- ✓ The admin is able to register one or more students and take their fingerprint template through scanning.
- ✓ The admin is able to add one or more sessions.
- ✓ Every course has a period which is made up of start time , ending time and the days concerned which makes up a session.
- ✓ A student can take attendance on placing his/her biometric .
- ✓ An admin can generate reports of the attendance taken by students.

These relations can be used to design an entity relationship among various entities

Fingerprint Student Attendance App ER-Diagram

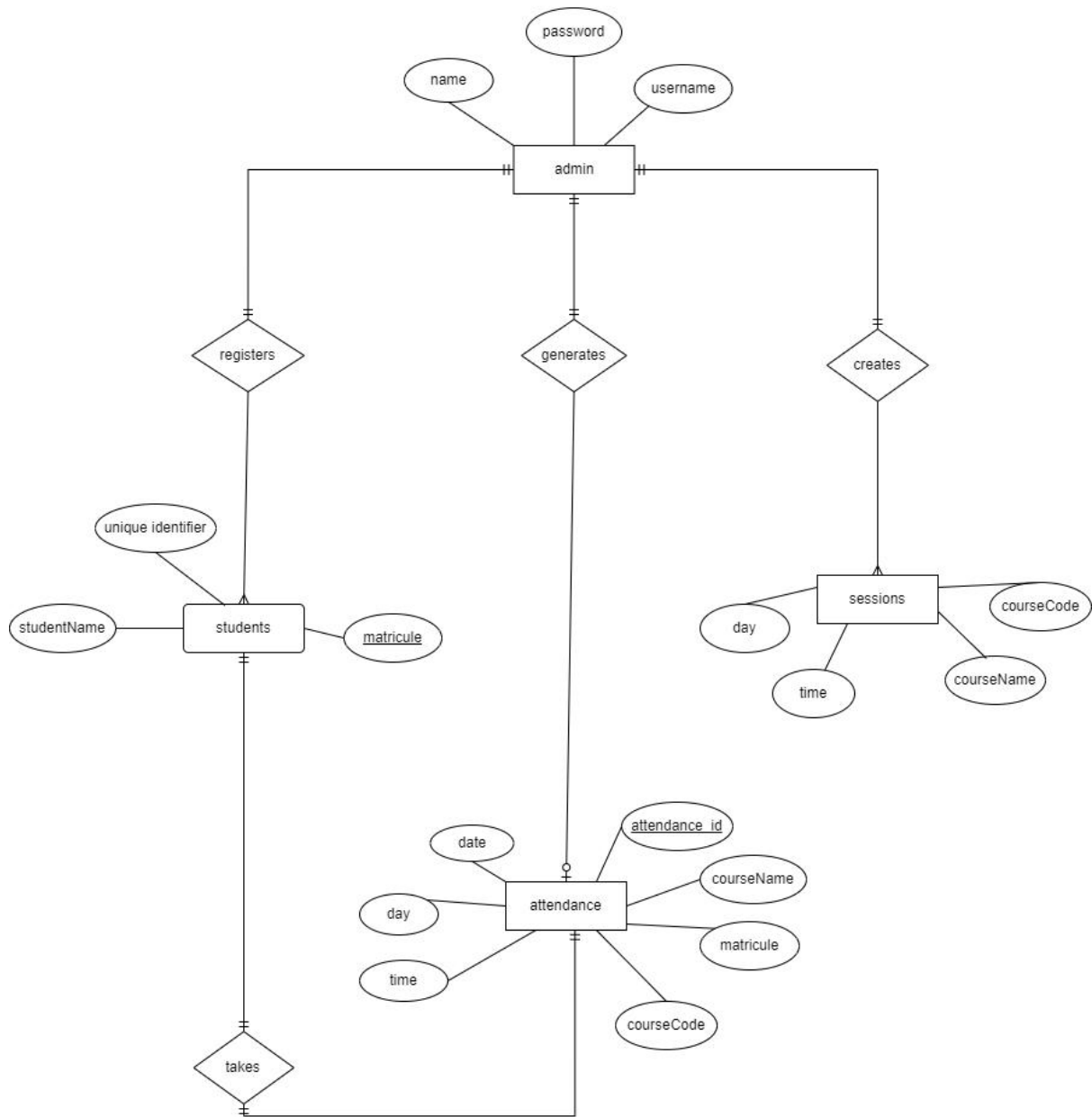


Figure 1 : Entity Relationship Diagram

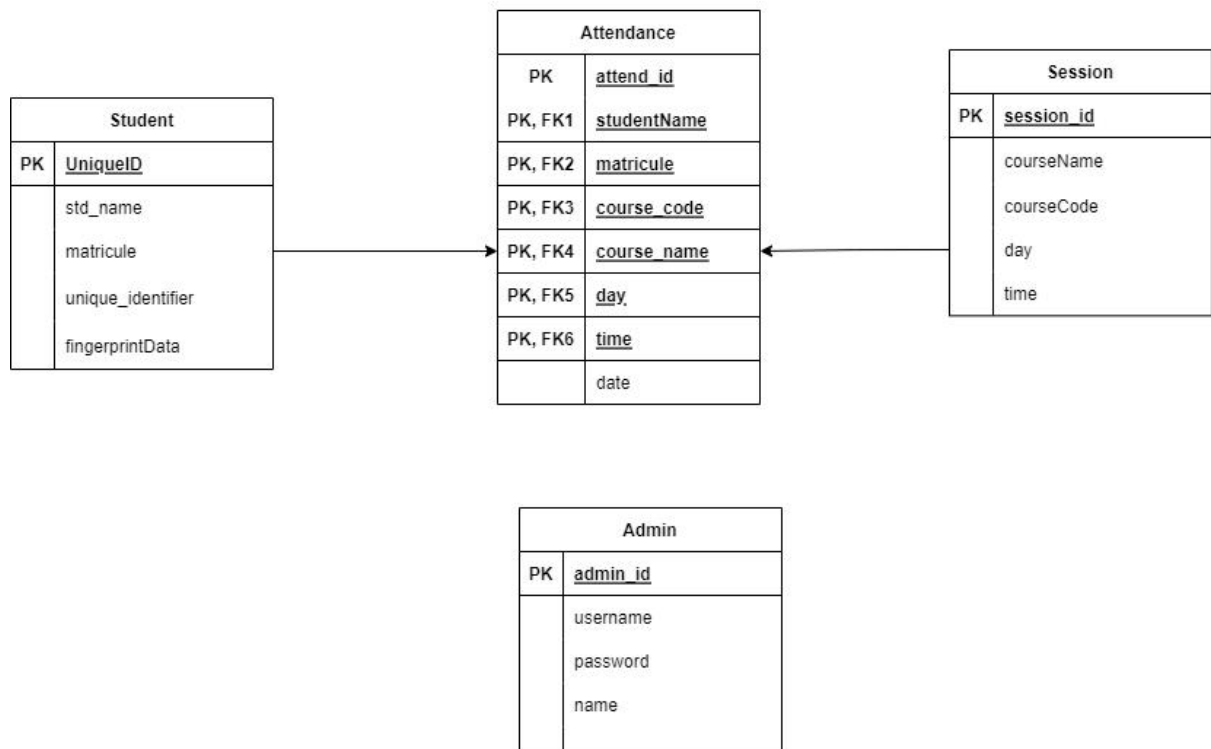
4. Relational Schema

The relational schema of this system shows all data interactions made by the various database tables (collections). It illustrates all entities and their atomic attributes in a normalized format.

The main entity in the relational schema is the Attendance entity which depends on the input data from the student collection and the session collection when the student in question selects the session to take his/her attendance generating attendance data.

The relational schema is as shown below

Fingerprint Student Attendance App Relational Database Schema

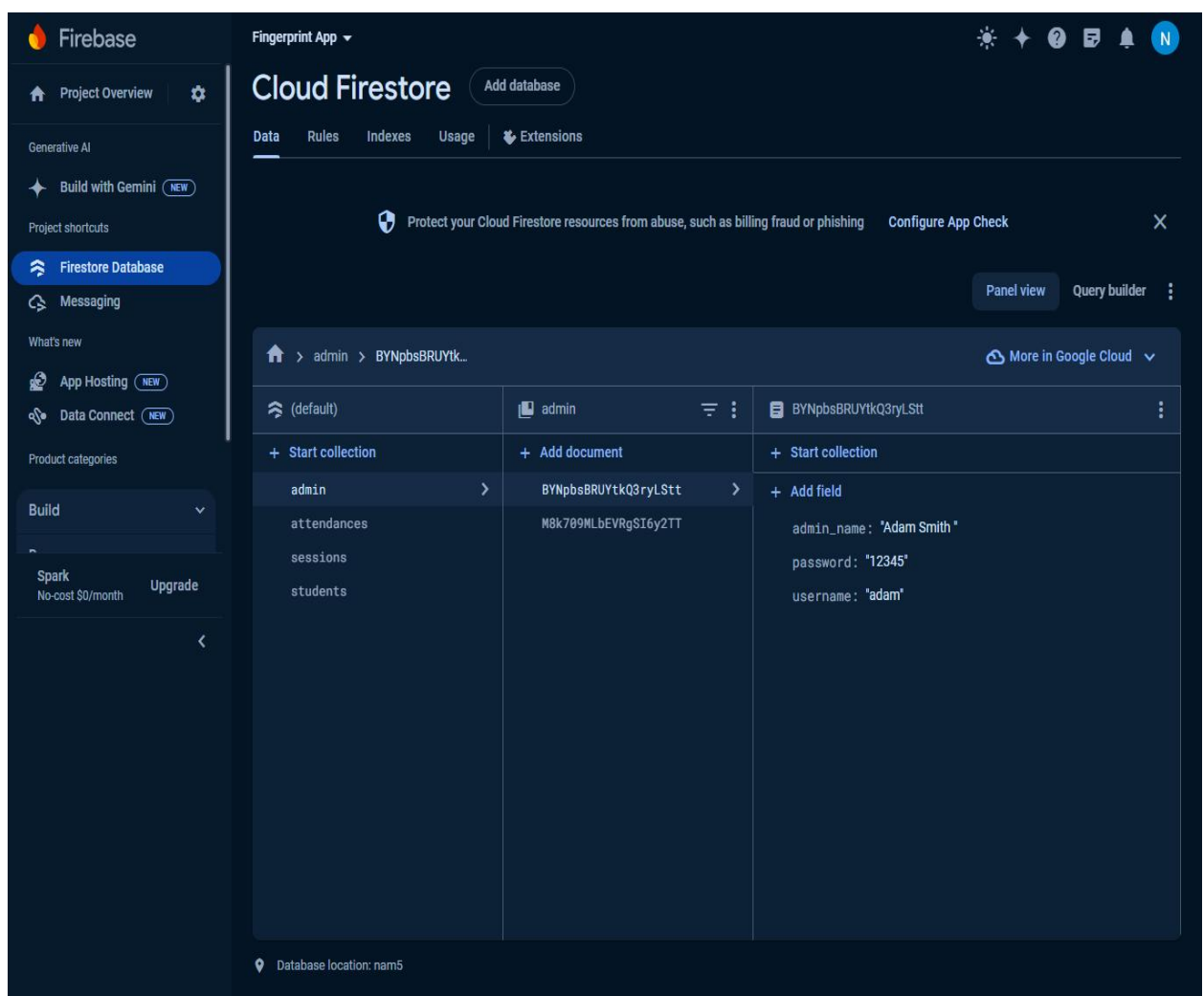


5. Firebase Implementation

The database for the fingerprint student attendance app has been implemented on Firebase using Firestore, the image implementations are shown below:-

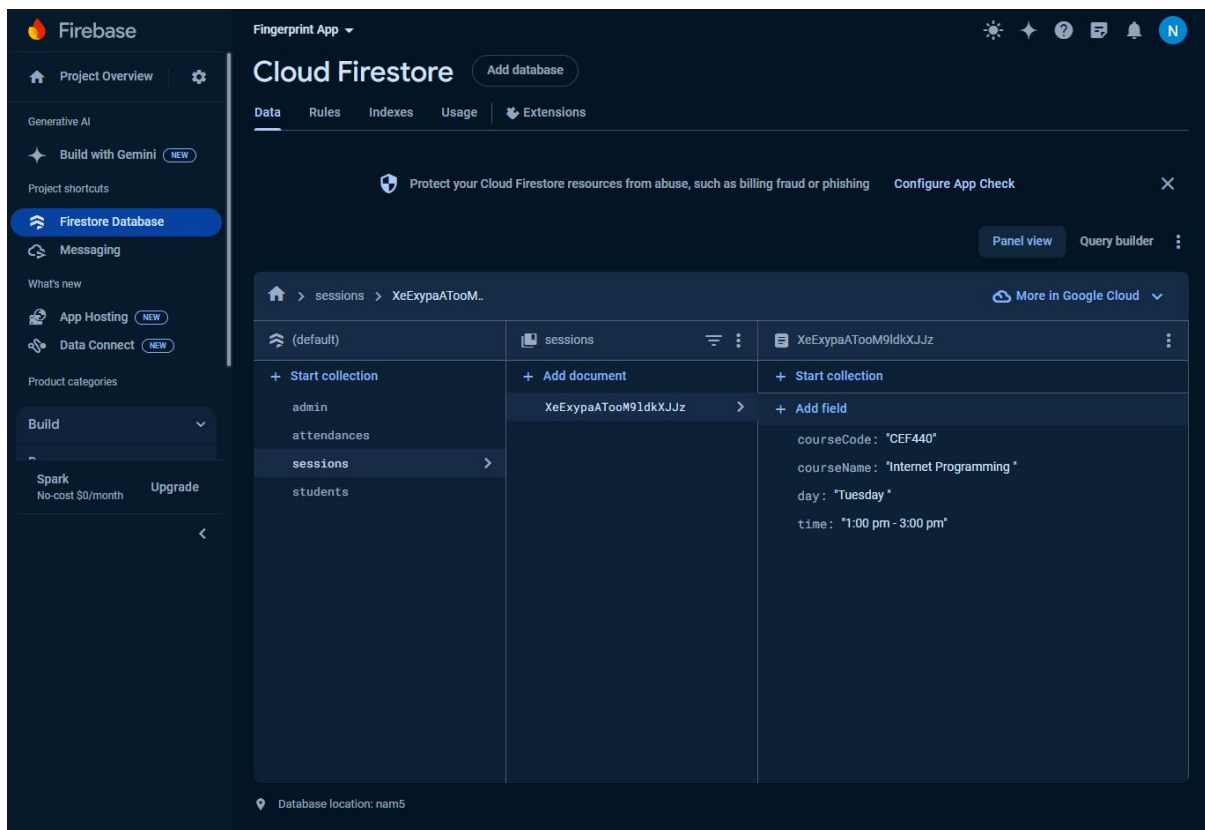
- ✓ Admin collection
- ✓ Student collection
- ✓ Session collection
- ✓ Attendances collection

5.1 Admin Collection



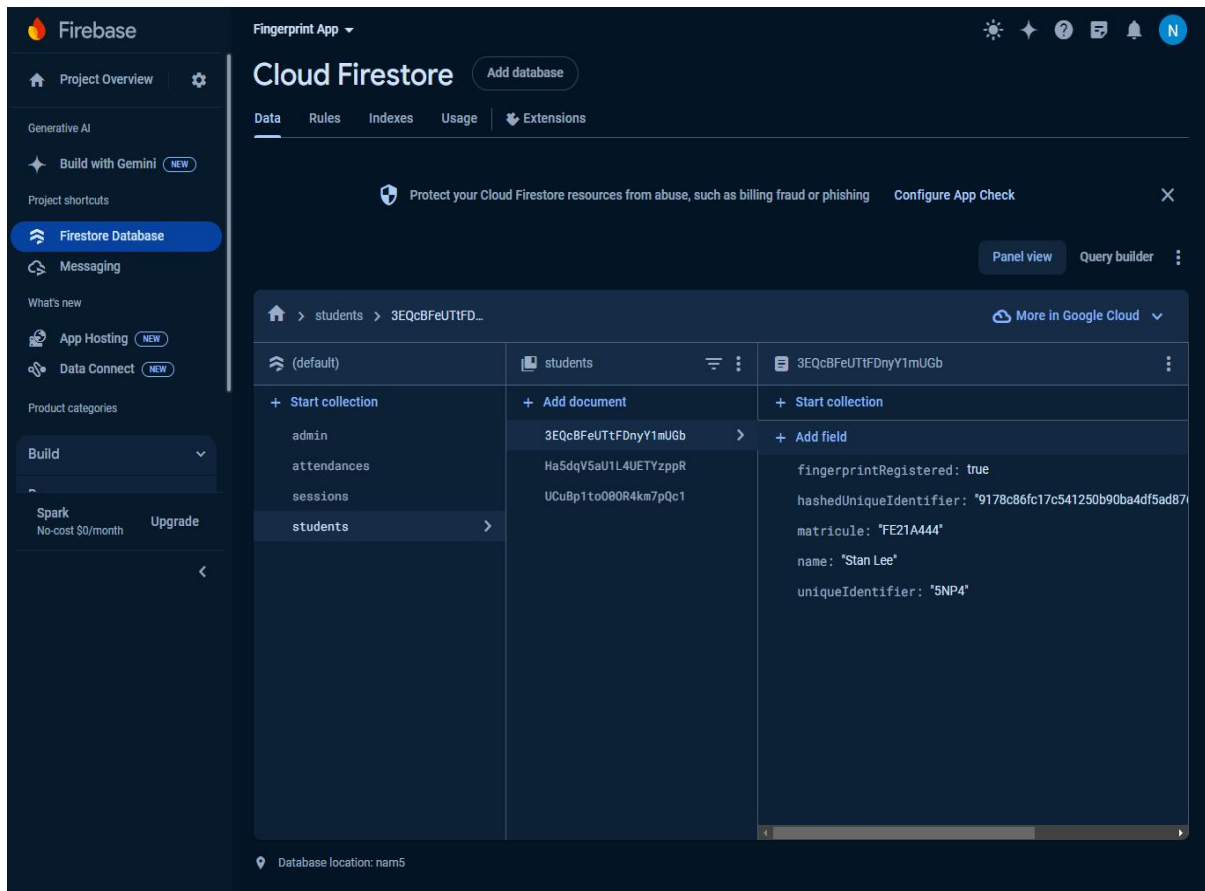
Admin Collection in Firestore

5.2 Student Collection



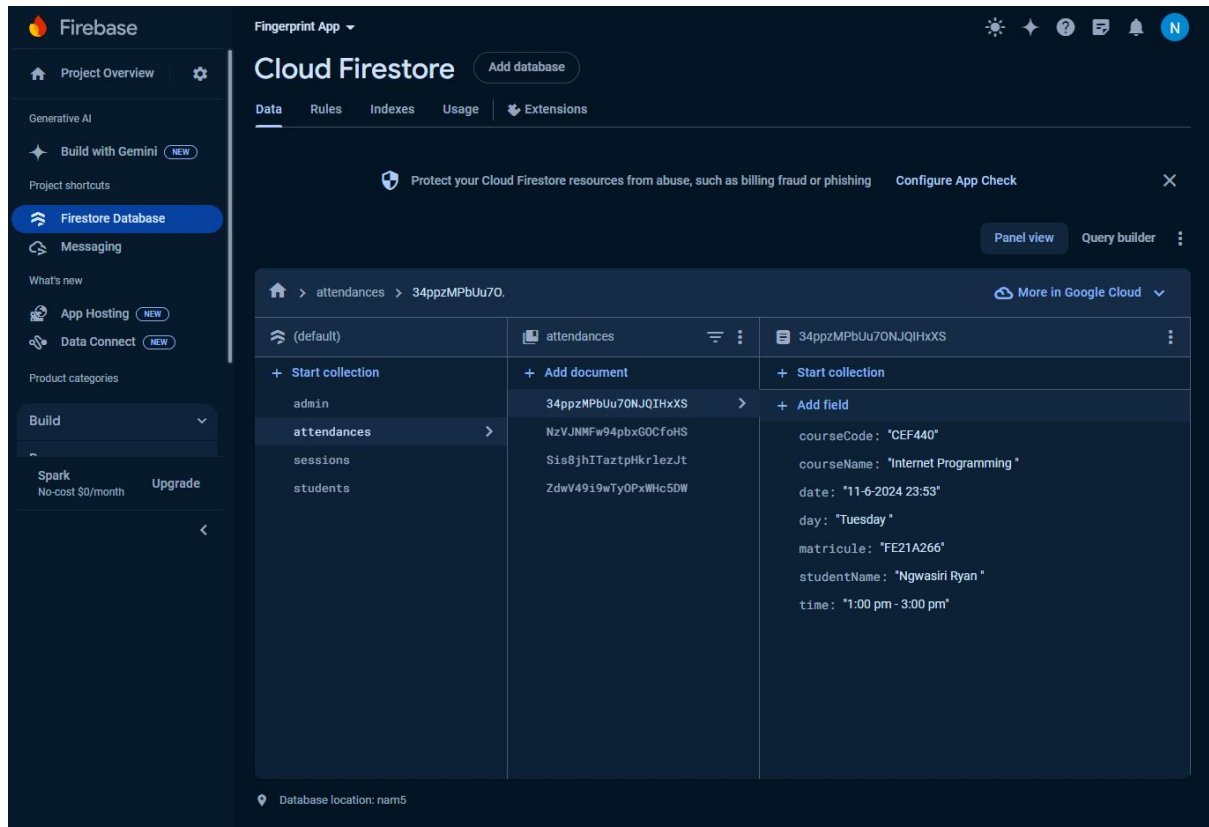
Attendances Collection in Firestore

5.3 Session Collection



Session Collection in Firestore

5.4 Attendances Collection



Attendances Collection in Firestore

6. Conclusion

In summary, the database design for the Attendance Report Generator app is focused on creating a robust, efficient, and scalable system for managing attendance records. By leveraging Firebase Firestore, the app benefits from real-time data synchronization, easy scalability, and secure data handling. This solid database foundation is crucial for providing users with a reliable and efficient tool for generating and managing attendance reports.

7. References

- ✓ Garcia-Molina, J. D. Ullman, and J. Widom, Database Systems: The Complete Book, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall, 2008.
- ✓ Chandra and A. Konak, "A multi-criteria decision-making approach for database design," in Proc. IEEE Symp. Computational Intelligence in Cyber Security (CICS), Nashville, TN, USA, 2009, pp. 37-44.
- ✓ Paulson, J. T. Haubert, and M. Stocker, "Evaluating NoSQL databases: A critical review of current practice," in Proc. IEEE 36th Int. Conf. Distributed Computing Systems Workshops (ICDCSW), Nara, Japan, 2016, pp. 25-32.
- ✓ Kumar, "Firebase and Firestore: Real-time database and cloud Firestore," in Proc. IEEE Int. Conf. Cloud Computing (CLOUD), San Francisco, CA, USA, 2018, pp. 180-185.
- ✓ Firebase, "Firestore | NoSQL Database | Google Cloud," [Online]. Available: <https://firebase.google.com/docs/firestore>. [Accessed: 12-Jun-2024].
- ✓ Doe, "An Overview of Firebase Firestore and Its Use in Mobile Applications," Technical Report, Google Inc., Mountain View, CA, USA, 2021.
- ✓ IEEE Standard for Database Design and Management, IEEE Std 1234-2020, IEEE Standards Association, 2020.