REPUBLIC OF CAMEROON
Peace-Work-Fatherland
MINISTER OF HIGHER EDUCATION
FACULTY OF ENGINEERING
AND TECHNOLOGY

REPUBLIQUE DU CAMEROON
PAIX-Travail-Patrie
MINISTRE DE L'ENSEIGNEMENT
SUPERIEUR
FACULTE D'INGINERIE
ET TECHGNOLOGIE

FACULTY OF ENGINEERING AND TECHNOLOGY
Department : Computer Engineering

CEF 440: Internet Programming and Mobile programming

TASK 6

DATABASE DESIGN AND IMPLEMENTATION

GOUP13

GROUP MEMBERS

Names	Matriculation
ALAIN NKEH MBUNKUH	FE21A133
ANANFACK ZANGO ANGELA CARINE	FE21A137
CHO WESLEY MUNGO	FE21A160
NKWETI MANGEM ANCELIA	FE21A280
NTUI RAOUL NTUI NJOCK	FE21A288

COURSE INSTRUCTOR : Dr NKEMENI VALERY

1. INTRODUCTION

In today's educational environment, ensuring accurate and efficient attendance tracking is a critical component for both administrative operations and academic integrity. Traditional methods of attendance tracking, such as manual roll-calling or paper-based sign-in sheets, are prone to errors, inefficiencies, and potential manipulation. To address these issues, a Biometric Student Attendance System (BSAS) offers a robust solution by leveraging biometric technology to automate and secure the process of student attendance recording.

This report presents a comprehensive database and implementation of a Biometric Student Attendance System. The primary objective of this system is to enhance the reliability and efficiency of attendance tracking through the use of biometric authentication methods, such as fingerprint or facial recognition. By implementing such a system, educational institutions can achieve a higher level of accuracy in attendance records, reduce administrative burdens, and enhance overall security.

2. Database Design Overview

Database design is the process of creating a detailed data model for a database. This involves defining the structure, storage, and retrieval mechanisms for the data to ensure it meets the requirements of its users and supports the application's functionality. The main goals of database design are to organize data in a way that reduces redundancy, maintains data integrity, and allows for efficient data retrieval and manipulation. Designing a database involves several critical steps to ensure it meets the needs of its users and performs efficiently. Here's an overview of the key steps in the database design process:

i. Requirement Analysis:

- Understand the requirements of the database users and the business needs.
- Identify the data to be stored, the relationships between data items, and the data retrieval needs.

ii. Conceptual Design:

- Create a high-level data model, typically an Entity-Relationship Diagram (ERD).
- Identify entities, attributes, and relationships.

- Define primary keys (unique identifiers for entities) and foreign keys (attributes that create relationships between entities).

iii. **Logical Design:**

- Convert the conceptual model into a logical model that can be implemented in a specific database management system (DBMS).
- Normalize the data model to eliminate redundancy and ensure data integrity.
- Define tables, columns, data types, and constraints (e.g., primary keys, foreign keys, unique constraints).

iv. **Implementation:**

- Create the database schema using a DBMS.
- Write SQL scripts to create tables, define relationships, and enforce constraints.
- Populate the database with initial data if necessary.

3. Database Design Process

3.1. Requirement Analysis

a. Functional Requirement

Our biometric attendance would typically consist of the following key components:

- **User Requirements:**
  - **Students**: Register and log attendance using biometric data**.**
  - **Lecturers/Administrators:** View attendance records, generate reports, and manages student's data.
  - **Administrators** also manages lecturers, and classes, assign lecturer to a specific course.

- **Biometric/fingerprint data storage:** Students biometric data or fingerprint is store in the database.
- **Attendance records and Report Generation:** Lecturers/Administrator view attendance records, and generate reports.
- **Manages Class Roster:** Lecturer manages class roster.

b. Data Requirement

- Student data: Name, Student_id, level name, Dept_name, email, institutional email, Faculty name.
- Attendance record date: Record_id/Session_id, Student_id, Course_id, time_slot_id, Record time
- Fingerprint data: fingerprint_id, student_id, fingerprint template.
- Course: Course_id, course name, level name, dept_name, credit value, Course status.

3.2. Conceptual Design

Conceptual design is an early phase of the database design process where you create a high-level representation of the data and its relationships without worrying about how it will be physically implemented in a database management system (DBMS). This phase focuses on understanding and modeling the domain requirements in a way that is easily understandable and independent of technical constraints. The main goal is to ensure that all necessary data and relationships are identified and accurately represented.

- ❖ **Key Elements of Conceptual Design**

a. Entities: Represent real-world objects, concepts, or events that you want to store information about. We came out with all the various entities for our database. Here there include:

- Students
- Instructors
- Administrators
- Courses
- Attendance Records
- Fingerprint Data
- Time slots
- Enroll
- Teaches
- Class
- Faculty
- Level
- Department
- Faculty

b. Attributes: Define the specific properties or characteristics of an entity. For example, a Student entity might have attributes like student ID, name, and email address

Student

Attributes:

Student ID (primary key)

Name

Institutional email

Alternative email

Faculty

Level

Faculty

Attributes:

Faculty ID (primary key)

Faculty Name

Building

Administrator

Attributes:

Admin ID (primary key)

Name

Position

Email

Permission

Course

Attributes:

Course ID (primary key)

Course Name

Department

Level

Credits

Status

Enrolment

Attributes:

Student ID (foreign key)

Course ID (foreign key)

Level Name

Semester

Year

Attendance Record

Attributes:

Record ID (primary key)

Student ID (foreign key)

Course ID (foreign key)

Attendance Status

Time Slot ID (foreign key)

Time Slot

Attributes:

Time Slot ID (primary key)

Day

Start Time

End Time

Fingerprint Data

Attributes:

Fingerprint ID (primary key)

Student ID (foreign key)

Fingerprint Template

Instructor

Attributes:

Instructor ID (Primary key)

Instructor Name

Email

Department

c. Relationships: Describe the connections or associations between entities. These connections can be one-to-one, one-to-many, or many-to-many.

- **Student Enrolls in Courses (Many-to-Many):** A student can enroll in many courses, and a course can have many students enrolled. This relationship is represented by the Enrollment entity, which has foreign keys for both Student ID and Course ID.
- **Administrator Manages Students (One-to-Many):** An admin can manage many students, but a student can only be managed by one admin. This relationship is implied by the fact that the Student entity does not have a foreign key for Admin ID.
- **Lecturer Teaches Courses (One-to-Many)**: A Lecturer can teach many courses, but a course can only be taught by one faculty member. This relationship is implied by the fact that the Course entity does not have a foreign key for Faculty ID.
- **Course Has Attendance Records (One-to-Many):** A course can have many attendance records, but an attendance record belongs to only one course. This relationship is represented by the foreign key Course ID in the Attendance Record entity.

- **Student Has Attendance Records (One-to-Many):** A student can have many attendance records, but an attendance record belongs to only one student. This relationship is represented by the foreign key Student ID in the Attendance Record entity.
- **Time Slot Defines Attendance Records (One-to-Many):** A time slot can define many attendance records, but an attendance record is defined by only one time slot. This relationship is represented by the foreign key Time Slot ID in the Attendance Record entity.
- **Student Has Fingerprint Data (One-to-One):** A student can have one fingerprint record, and a fingerprint record belongs to only one student. This relationship is represented by the foreign key Student ID in the Fingerprint Data entity.

d. ER Diagram: A visual representation of the entities, attributes, and relationships.

Components:

- **Entities**: Represented as rectangles.
- **Attributes**: Represented as ovals connected to their respective entities.
- **Relationships**: Represented as diamonds or lines connecting entities.

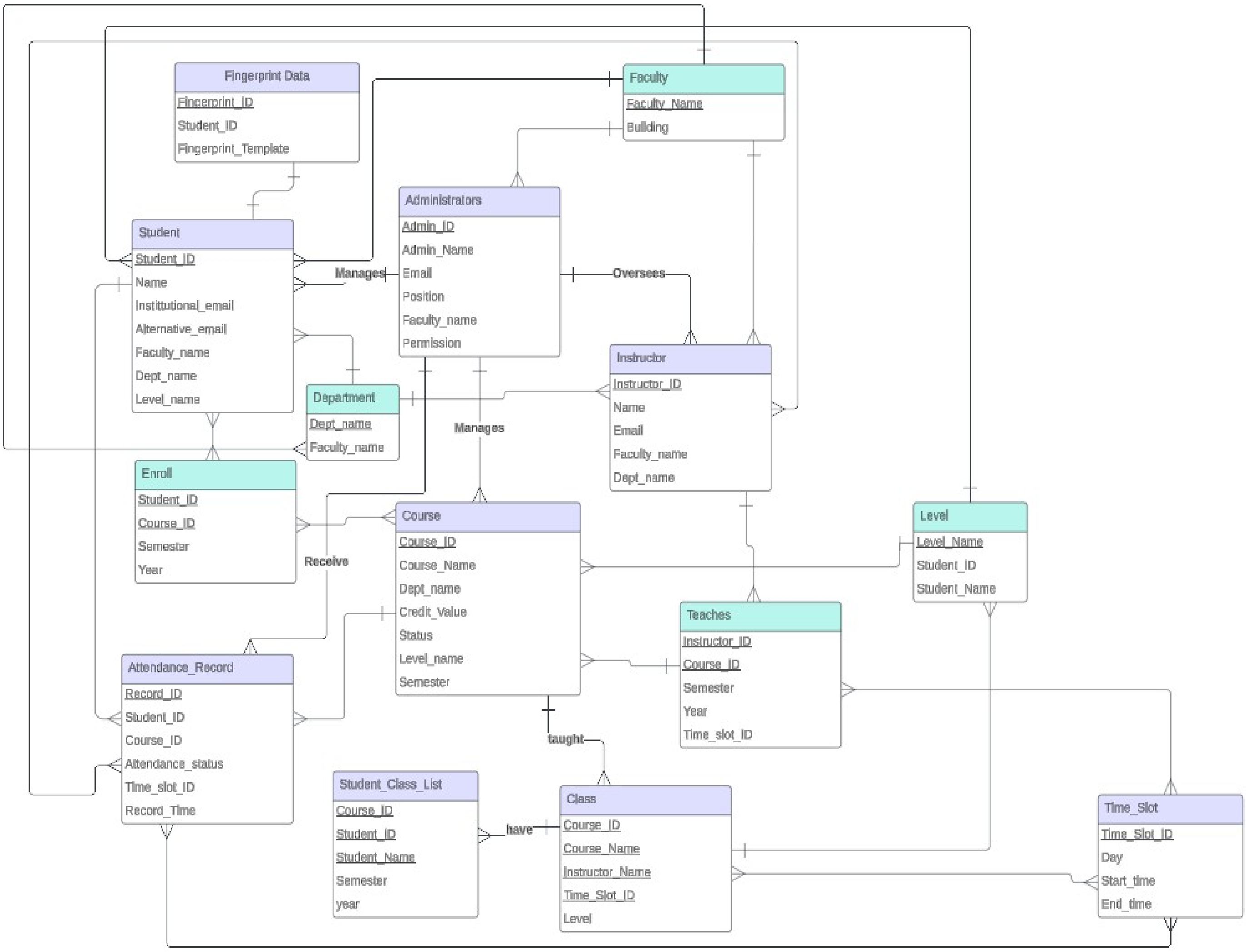


Figure 1: ER Diagram of BioCheck-In

3.3. Logical Design

Logical design is the phase in the database design process where the conceptual model is transformed into a logical data model that can be implemented in a specific database management system (DBMS). This stage focuses on defining the structure of the database in a way that is independent of any particular DBMS but detailed enough to be converted into a physical design.

3.4. Database Implementation with Firebase Real-Time Database

Database implementation involves translating the database designs into an actual database using a Database Management System (DBMS). This process includes creating the database schema, populating it with data, and setting up the necessary environment for its operation**.**

3.4.1. Reason for Chosen Firebase Real-Time Database as our DMS

Choosing Firebase Real-time Database for a biometric student attendance mobile application can offer several benefits that align well with the requirements of such an application. Here are some reasons why Firebase Real-time Database is a suitable choice:

- **. Real-time Data Synchronization**

  - **Instant Updates:** Firebase Real-time Database provides real-time data synchronization across all connected devices. This feature is crucial for an attendance system, where immediate updates are needed to reflect students' attendance status accurately.

  - Live Data: Changes made in the database are instantly reflected in the app, ensuring that attendance records are always up-to-date.

- **Scalability**

  - Handle Large User Base: Firebase can scale effortlessly to handle a large number of students and attendance records, accommodating the growth of the institution.

  - Automatic Scaling: Firebase automatically scales the infrastructure to handle increased loads, ensuring consistent performance without manual intervention.

- **Offline Capabilities**

  - **Offline Access:** The Firebase Real-time Database supports offline data access. This is essential for a mobile application, allowing students and teachers to mark attendance even without an internet connection. The data is synchronized once the connection is restored.

  - **Local Caching**: Data is cached locally on the device, which ensures the app remains functional even in environments with intermittent connectivity.

- **Security**

  - **Firebase Authentication:** Integrating Firebase Authentication provides secure user authentication mechanisms. This is particularly important for biometric applications to ensure that only authorized users can access and modify attendance records.

  - Data Security: Firebase provides security rules that can be configured to control access to the data, ensuring that sensitive information is protected.

- **Ease of Integration**

  - Biometric Integration: Firebase provides seamless integration with various authentication methods, including biometric systems. This makes it easier to incorporate biometric attendance tracking.

  - SDKs and Libraries: Firebase offers SDKs for multiple platforms (iOS, Android, Web), simplifying the development process and reducing the time to market.

- **Easy Deployment and Maintenance**

  - Quick Setup: Firebase allows for quick setup and deployment, enabling rapid development and deployment cycles.

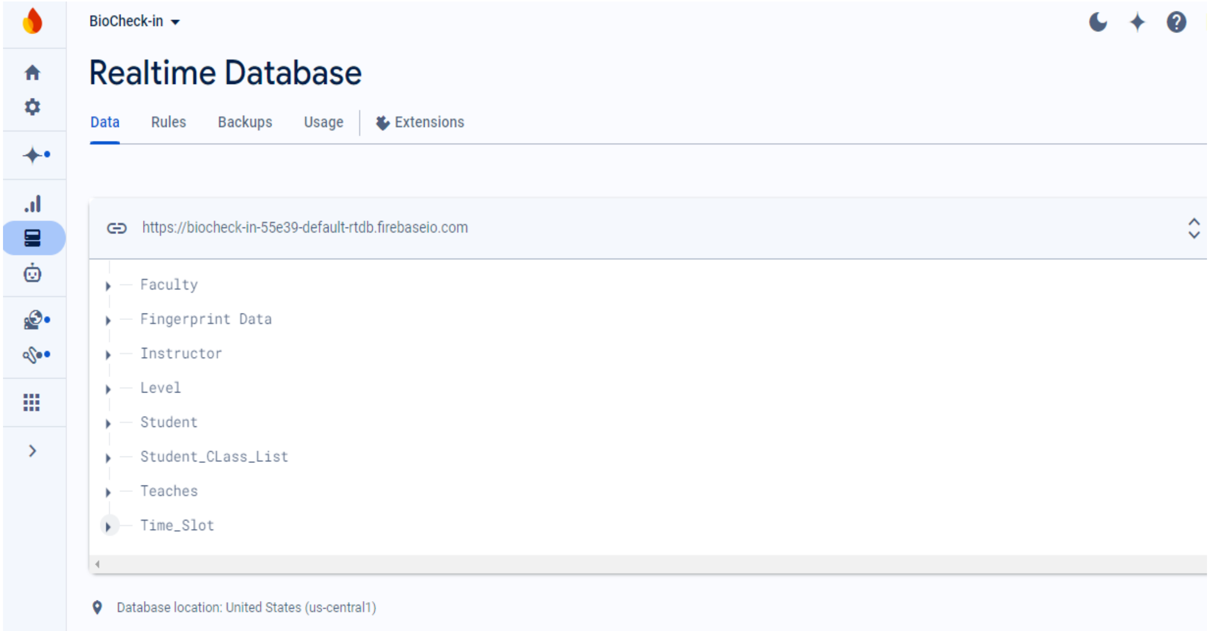
  - Automatic Updates: Firebase handles database updates and maintenance, ensuring that the app runs on the latest and most secure version without manual intervention.

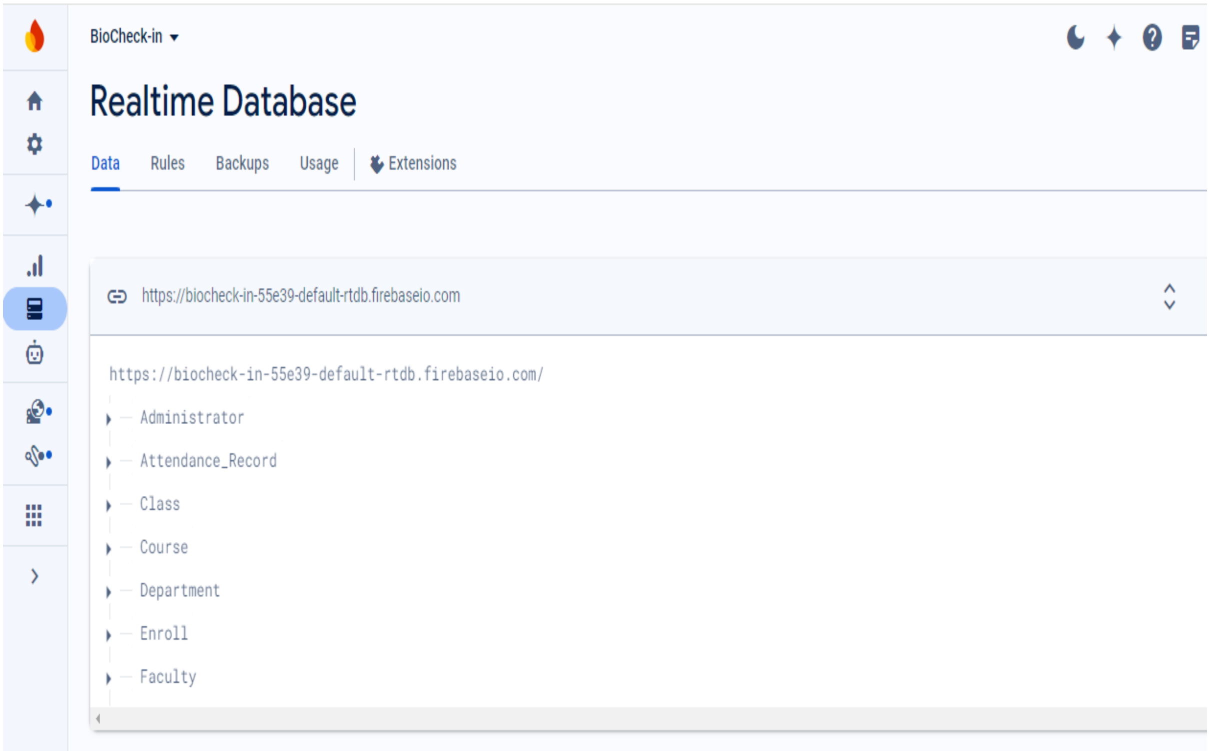
  .

3.4.2. Implementation Phase

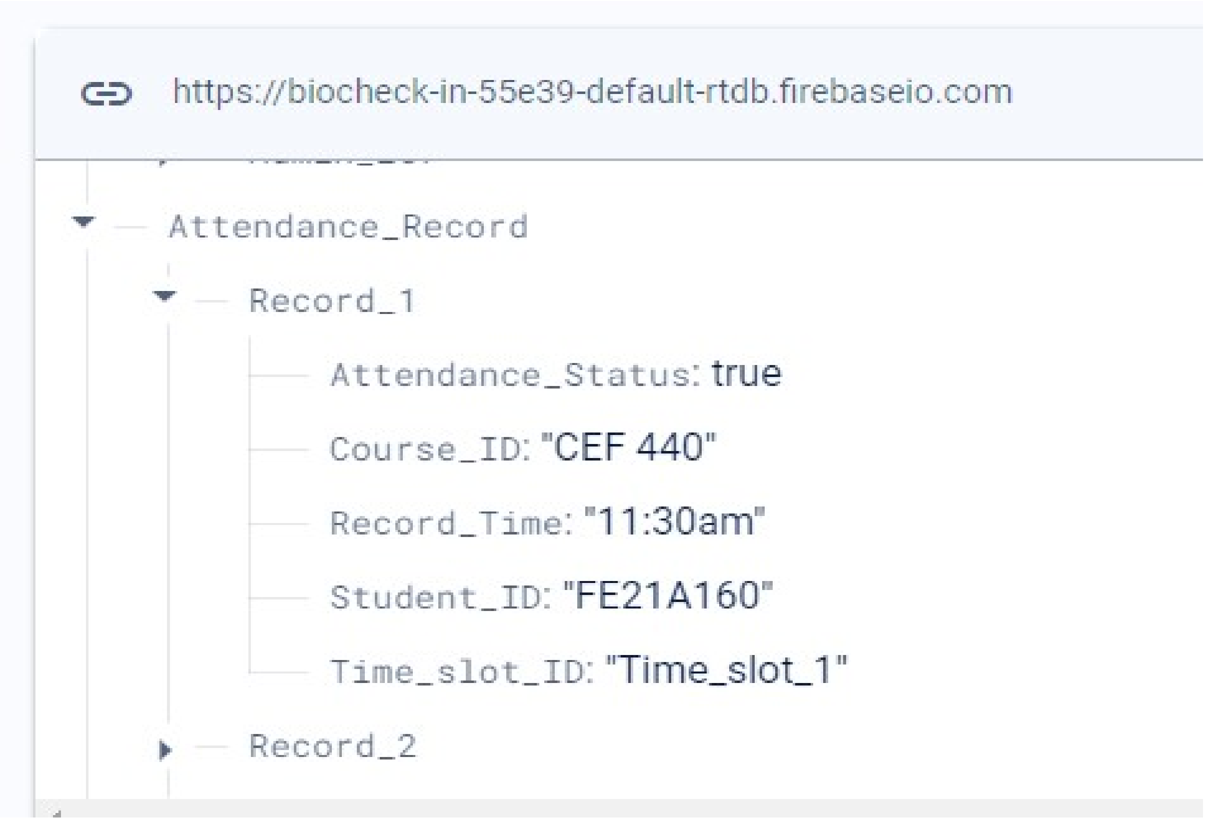
We created a firebase project and configure Firebase Real-time Database, define the data structure considering of collections and documents for entities like students, lecturers, Attendance record, Classes, etc. and finally we implemented the Data model that is create collections and documents using firebase.

All the various collections and documents for various entities in our database that we created in Firebase Real-Time Database





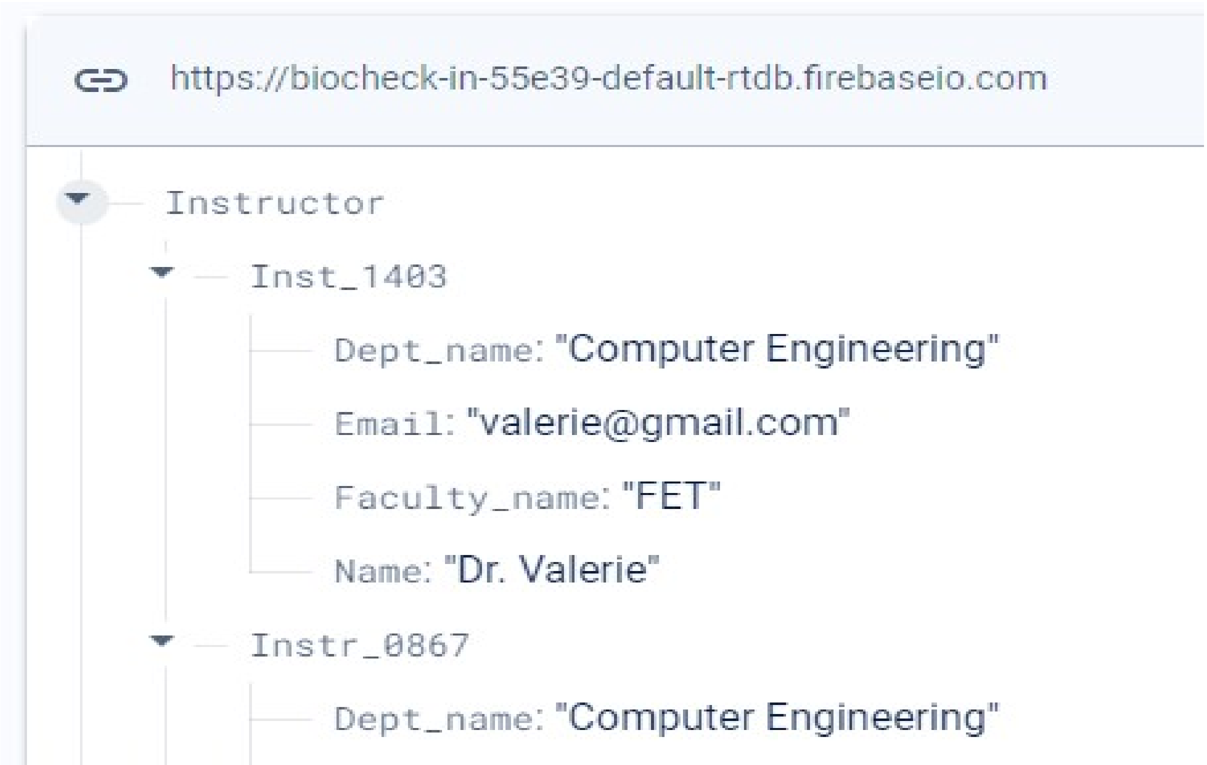
➢ **Administrator Collection**



- **Attendance Records Collection**

- **Student Collections**

➢ **Instructor Collection**



4. Conclusion

Database implementation is a critical phase where the theoretical design is transformed into a functional system. Following a structured approach ensures that the database is robust, secure, and performs well under real-world conditions.