****

**REPUBLIQUE DU CAMEROON**

**PAIX-Travail-Patrie**

**MINISTRE DE L’ENSEIGNEMENT SUPERIEUR**

**FACULTE D’INGINERIE**

**ET TECHGNOLOGIE**

**REPUBLIC OF CAMEROON**

**Peace-Work-Fatherland**

**MINISTER OF HIGHER EDUCATION**

**FACULTY OF ENGINEERING**

**AND TECHNOLOGY**

***\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\** UNIVERSITY OF BUEA *\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\****

**CEF 440: INTERNET PROGRAMMING AND MOBILE PROGRAMMING**

**Task 6: Database Design and Implementation**

**PRESENTED BY:**

|  |  |
| --- | --- |
| **NAME** | **REGISTRATION NUMBER** |
| **IHIMBRU ZADOLF ONGUM** | **FE21A203** |
| **CHE KASSINA KUM** | **FE21A158** |
| **NFOUA EUGENE MGBA** | **FE21A257** |
| **FONJI DANIEL KUKUH** | **FE21A194** |
| **EPIE MUKEH SANDRA** | **FE21A185** |

**COURSE FACILITATOR: JUNE 2024**

**Dr. VALERY NKEMENI Group 12**

Table of Contents

**Introduction3**

**System Architecture3**

**Database Design 4**

Requirement Analysis4

Entities and Attributes4

Entity Relationship Diagram5

Normalization6

**DBMS Selection and Reasons 7**

**Database Schema 9**

**Conclusion 11**

1. **Introduction**

The Biometric Student Attendance Recording Mobile Application (i-tend) is designed to automate and streamline the attendance tracking process in educational institutions using biometric technology. This application ensures accurate and secure attendance recording by leveraging fingerprint recognition. It is intended for use by lecturers and administrators, who can initiate and manage attendance sessions, register students and teachers, and generate reports at the end of sessions.

#### **Project Scope**

The scope of this project includes:

1. **User Management:**
   * Registration of students and teachers by administrators.
   * Authentication for lecturers and administrators.
2. **Attendance Management:**
   * Initiating attendance sessions by lecturers.
   * Marking attendance using fingerprint recognition.
   * Viewing attendance records and generating reports.
3. **Notification System:**
   * Displaying alerts and messages to users.
4. **Database Management:**
   * Storing user data, attendance records, and biometric data.
   * Ensuring data integrity and security.
5. **System Architecture**

The system architecture for the biometric attendance application includes the following components:

***Frontend (Mobile Application)***

- Developed using React Native

- Features include user authentication, attendance marking, and report generation

***Backend (Server)***

- Developed using Node.js and Express.js

- Handles API requests from the mobile application

- Manages database interactions and biometric data processing

***Database***

- Firebase for storing user data, attendance records, and session information

***Biometric Hardware***

- Fingerprint scanners integrated with mobile devices for biometric data capture

1. **Database Design**

#### **3.1 Requirements Analysis**

1. **Functional Requirements**

**Home:** Returns to the main dashboard screen.

**Profile:** Access to teacher's profile settings and information.

**Notifications:** Displays alerts and messages.

**Logout:** Log out of the system.

**For Lecturer:**

* Initiate Attendance
* View Attendance
* Mark Attendance
* Generate Report
* Add Session
* View Session

**For Administrator:**

* Register Student
* Register Teacher
* View Attendance
* Generate Report
* View Student
* Add Session
* Add Administrator
* View Session

1. **Non-Functional Requirements**

**Security:** Ensure data security and user privacy by implementing encryption and secure access protocols.

**Performance:** The system should be able to handle multiple requests simultaneously and provide real-time responses.

**Scalability:** The system should be scalable to accommodate more users and data in the future.

**Usability:** The application should have an intuitive interface that is easy to navigate for all users.

**Compatibility:** The application should be compatible with various mobile devices and biometric hardware.

#### **3.2 Entities and Attributes**

The main entities and their attributes in the database are:

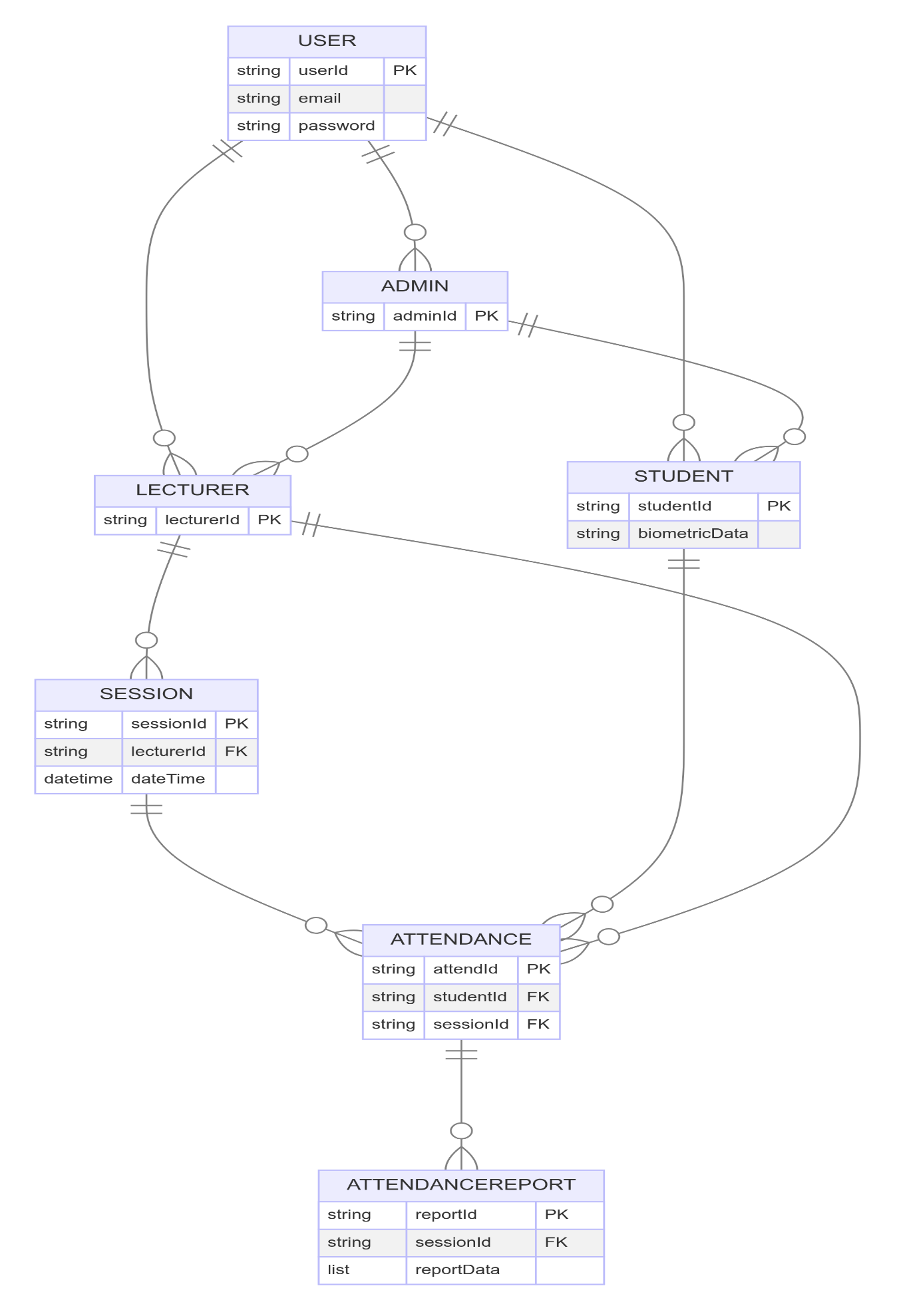
1. **User (Base Entity)**
   1. **Attributes:**
      1. userId (Primary Key)
      2. email
      3. password
   2. **Relationships:**
      1. Inherited by Admin, Lecturer, and Student.
2. **Admin (Inherits from User)**
   1. **Attributes:**
      1. adminId (Primary Key)
   2. **Relationships:**
      1. Manages multiple Students.
      2. Manages multiple Lecturers.
3. **Lecturer (Inherits from User)**
   1. **Attributes:**
      1. lecturerId (Primary Key)
   2. **Relationships:**
      1. Manages multiple Sessions.
      2. Marks multiple Attendance records.
4. **Student (Inherits from User)**
   1. **Attributes:**
      1. studentId (Primary Key)
      2. biometricData
   2. **Relationships:**
      1. Makes multiple Attendance records.
5. **Session**
   1. **Attributes:**
      1. sessionId (Primary Key)
      2. cturleerId (Foreign Key)
      3. dateTime
   2. **Relationships:**
      1. Contains multiple Attendance records.
      2. Managed by one Lecturer.
6. **Attendance**
   1. **Attributes:**
      1. attendId (Primary Key)
      2. studentId (Foreign Key)
      3. sessionId (Foreign Key)
   2. **Relationships:**
      1. Belongs to one Session.
      2. Made by one Student.
7. **AttendanceReport**
   1. **Attributes:**
      1. reportId (Primary Key)
      2. sessionId (Foreign Key)
      3. reportData (List of Attendance)
   2. **Relationships:**
      1. Contains multiple Attendance records.

#### **3.3 Entity Relationship Diagram**

The ER (Entity-Relationship) diagram for the Biometric Attendance System following the Entities and Attributes listed above, as well as the Relationships below, is as follows:

**Relationships**

1. **User-Admin, User-Lecturer, User-Student:** Inheritance relationship: Admin, Lecturer, and Student inherit from User.
2. **Admin-Student, Admin-Lecturer:** One-to-many aggregation: Admin manages multiple Students and Lecturers.
3. **Lecturer-Session:** One-to-many aggregation: Lecturer manages multiple Sessions.
4. **Session-Attendance:** One-to-many association: Session contains multiple Attendance records.
5. **Student-Attendance:** One-to-many association: Student makes multiple Attendance records.
6. **Lecturer-Attendance:** One-to-many association: Lecturer marks Attendance for multiple students.
7. **Attendance-AttendanceReport:** One-to-many association: AttendanceReport contains multiple Attendance records.



#### **3.4 Normalization**

Normalization is the process of organizing data to minimize redundancy. For this database:

* First Normal Form (1NF): All tables have unique primary keys, and each column contains atomic values.
* Second Normal Form (2NF): All non-key attributes are fully functional dependent on the primary key.
* Third Normal Form (3NF): All attributes are non-transitive, meaning there are no dependencies between non-key attributes.

1. **DBMS Selection, and Reasons**

In our biometric student attendance tracking application, we chose Firebase as our Database Management System (DBMS).

1. **What is Firebase?**

.

Firebase is a BAAS product of Google that helps developers to build, manage, and grow their apps easily. Firebase provides a variety of services and features that enable developers to create dynamic web and mobile apps without worrying about the backend infrastructure which fits perfectly for our application.

1. **Why Firebase?**

* **Real-time Database**

Firebase provides a real-time database, which means any changes in the data are immediately reflected across all clients. This is crucial for our application as it ensures that attendance data is updated instantly, allowing teachers and administrators to see the latest information without delay.

* **Massive Open Source Documentation to Ease Learning**

Firebase comes with extensive open-source documentation that is incredibly beneficial for developers. This documentation is comprehensive, well-organized, and constantly updated, making it an excellent resource for both beginners and experienced developers.

* **Easy Integration**

Firebase is easy to integrate with mobile applications. Since our attendance system needs to work seamlessly across different platforms, Firebase’s cross-platform capabilities are a significant advantage.

* **Scalability**

As our application grows, we need a database that can scale without extensive reconfiguration. Firebase handles large amounts of data and many simultaneous connections gracefully, making it a future-proof choice for our application.

* **Security**

Security is a top priority for our application since it deals with sensitive student data. Firebase offers robust security features, including user authentication and data encryption. This ensures that only authorized users can access or modify the attendance records.

* **Offline Capabilities**

With Firebase, our application can work offline, syncing data once the device reconnects to the internet. This is beneficial for users who may have intermittent internet access, ensuring they can still use the application effectively.

* **Cost-Effective**

Firebase’s pricing model is flexible and cost-effective, especially for startups and growing applications. We can start with the free tier and upgrade as our needs expand, making it a budget-friendly option.

1. **Features of Firebase Used in Our Application**

* **Firestore.**

Firestore is Firebase’s NoSQL database, which stores data in documents organized into collections. It’s suitable for our attendance data as it allows for flexible, hierarchical data structures.

* **Authentication**

Firebase Authentication provides easy and secure authentication methods, including email/password and third-party logins (Google, Facebook, etc.). This is used to manage user access to the attendance tracking system.

* **Cloud Functions**

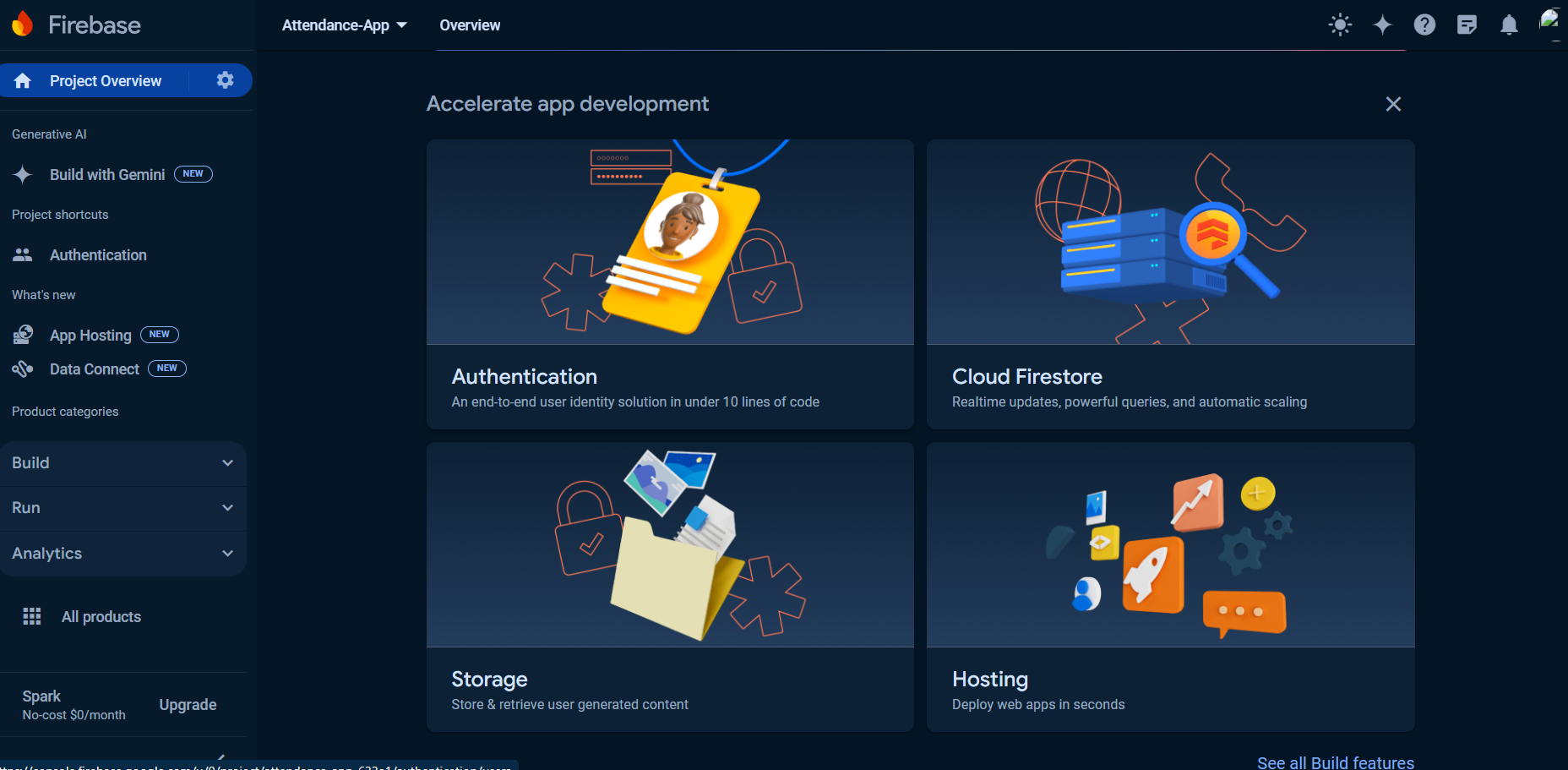
Cloud Functions for Firebase lets us run backend code in response to events triggered by Firebase features. This is useful for automating tasks such as sending notifications when a student checks in or out.

* **Cloud Storage**

Firebase Cloud Storage is used to store and serve user-generated content, such as profile pictures or scanned documents, in a scalable and secure way.

* **Analytics**

Firebase Analytics helps us understand user behavior and improve the application by providing insights into how the app is used, which features are popular, and where users are experiencing issues.



   Firebase is an excellent choice for our biometric student attendance tracking application due to its real-time capabilities, ease of integration, scalability, security, offline functionality, and cost-effectiveness. By leveraging Firebase's features, we can ensure our application is robust, responsive, and ready to grow with our user needs.

1. **Database Schema**

The database schema includes tables for students, lecturers, administrators, courses, sessions, attendance, and fingerprints. Below is the detailed schema:

**Students**

* `student\_id` (Primary Key, INT, AUTO\_INCREMENT)
* `first\_name` (VARCHAR(50))
* `last\_name` (VARCHAR(50))
* `email` (VARCHAR(100), UNIQUE)

**Lecturers**

* `lecturer\_id` (Primary Key, INT, AUTO\_INCREMENT)
* `first\_name` (VARCHAR(50))
* `last\_name` (VARCHAR(50))
* `email` (VARCHAR(100), UNIQUE)
* `password` (VARCHAR(10))

**Administrators**

* `admin\_id` (Primary Key, INT, AUTO\_INCREMENT)
* `first\_name` (VARCHAR(50))
* `last\_name` (VARCHAR(50))
* `email` (VARCHAR(100), UNIQUE)
* `password` (VARCHAR(10))

**Courses**

* `course\_id` (Primary Key, INT, AUTO\_INCREMENT)
* `course\_name` (VARCHAR(100))
* `course\_code` (VARCHAR(10), UNIQUE)
* `lecturer\_id` (Foreign Key, INT)

**Sessions**

* `session\_id` (Primary Key, INT, AUTO\_INCREMENT)
* `course\_name` (VARCHAR(100))
* `course\_code` (VARCHAR(10), UNIQUE)
* `lecturer\_id` (Foreign Key, INT)
* `start\_time` (TIME)
* `end\_time` (TIME)

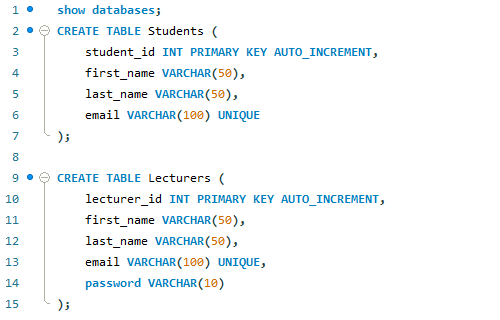
**Attendance**

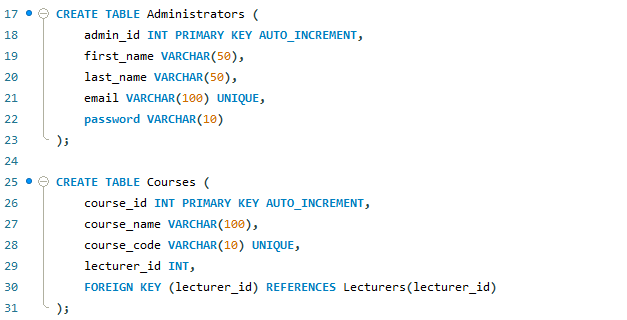
* `attendance\_id` (Primary Key, INT, AUTO\_INCREMENT)
* `student\_id` (Foreign Key, INT)
* `session\_id` (Foreign Key, INT)
* `status` (ENUM('Present', 'Absent'))
* `timestamp` (TIMESTAMP)

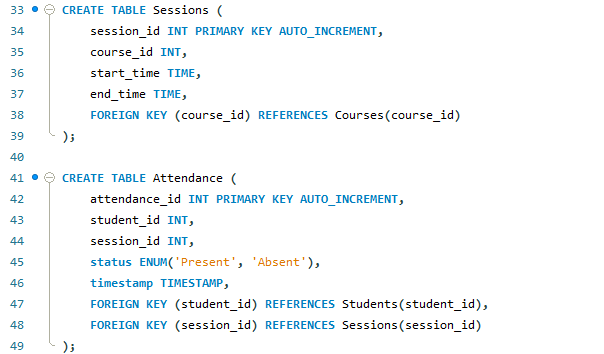
**Fingerprints**

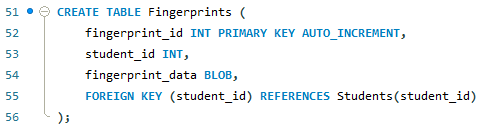
* `fingerprint\_id` (Primary Key, INT, AUTO\_INCREMENT)
* `student\_id` (Foreign Key, INT)
* `fingerprint\_data` (BLOB)

**SQL Scripts:**

****







1. **Conclusion**

The database design and implementation phase of the Biometric Student Attendance System was a critical component in the development process. This phase ensured that the application's data storage, retrieval, and management were robust, efficient, and secure. The following points summarize the key outcomes and achievements of this phase.

#### **Key Achievements:**

1. **Comprehensive Schema Design:**
   * A well-structured database schema was created, covering all essential entities such as Users, Students, Teachers, Courses, Sessions, Attendance, and Notifications.
   * The schema ensures data integrity and logical relationships between different entities, facilitating seamless data operations.
2. **Normalization:**
   * The database was normalized to the third normal form (3NF), which minimizes redundancy and ensures that data dependencies are correctly enforced. This leads to efficient storage and retrieval of data, reducing the potential for anomalies.
3. **Security Measures:**
   * Sensitive data, including passwords and biometric data, are securely stored using encryption techniques.
   * Role-based access control mechanisms are in place to ensure that only authorized users can perform specific operations, enhancing data security.
4. **Performance Optimization:**
   * Indexing strategies were applied to key columns to speed up query performance, ensuring quick access to frequently requested data.
   * The database design supports high concurrency, allowing multiple users to perform operations simultaneously without performance degradation.
5. **Scalability:**
   * The database was designed with scalability in mind, enabling it to handle increasing volumes of data and a growing number of users without compromising performance.
   * This scalability ensures that the system can accommodate future growth and additional features.
6. **Data Integrity:**
   * Foreign key constraints and other integrity rules were enforced to maintain consistent and accurate data across all tables.
   * Regular integrity checks and validation rules ensure that the data remains reliable and trustworthy.

#### **Challenges and Solutions:**

1. **Biometric Data Management:**
   * Storing large volumes of biometric data posed challenges in terms of storage and retrieval efficiency.
   * Solution: The use of Binary Large Objects (BLOBs) for storing fingerprint data and ensuring it is encrypted for security purposes.
2. **Maintaining Data Integrity:**
   * Ensuring the accuracy and consistency of data across multiple related tables required careful planning.
   * Solution: The use of foreign key constraints and transaction management to maintain referential integrity and prevent data anomalies.
3. **Performance Optimization:**
   * Optimizing the database for high performance was essential to handle real-time attendance tracking and reporting.
   * Solution: Implementing indexing on frequently accessed columns and optimizing SQL queries for better performance.

#### **Future Enhancements:**

1. **Advanced Analytics:**
   * Integrating advanced analytics and reporting tools to provide deeper insights into attendance patterns and trends.
2. **Cloud Integration:**
   * Exploring cloud-based database solutions to leverage benefits such as scalability, high availability, and disaster recovery.
3. **Real-time Data Processing:**
   * Enhancing the database to support real-time data processing for instant updates and notifications.
4. **Continuous Improvement:**
   * Regular updates and improvements based on user feedback and technological advancements to ensure the database remains robust and efficient.

### **Final Thoughts:**

The successful completion of the database design and implementation phase is a significant milestone in the development of our Biometric Student Attendance Recording Mobile Application ***(i-tend).*** The carefully designed database ensures efficient data management, high performance, and strong security, providing a solid foundation for the entire application. This phase's achievements not only meet the current requirements but also position the system for future enhancements and scalability, ensuring its long-term success and reliability.

1. **References**

* Connolly, T., & Begg, C. (2014). *Database Systems: A Practical Approach to Design, Implementation, and Management* (6th ed.). Pearson Education Limited.
* Elmasri, R., & Navathe, S. B. (2015). *Fundamentals of Database Systems* (7th ed.). Pearson.