### CPEN 208

11014727

PROJECT 1

ANNAN CHIOMA PRAISE

### Part 1: Database Development

#### 1. Introduction

The Computer Engineering Department aims to develop software to manage various administrative tasks such as handling student personal information, fees payments, course enrollments, and lecturer assignments. This section of the report details the design and implementation of a PostgreSQL database to support these functionalities.

#### 2. Database Design

The database includes tables for storing information about students, fees payments, courses, enrollments, lecturers, and teaching assistants. The database design ensures relational integrity and efficient data retrieval.

#### 3. Schema and Table Creation

##### Student Personal Information

This table stores basic personal information about the students.

* **Columns:** student\_id , first\_name, last\_name, date\_of\_birth, email.

##### Course Enrollment

This table contains information about the courses offered and the enrollments of students in these courses.

* **Courses Table Columns:** course\_id (Primary Key), course\_name, course\_code.
* **Enrollments Table Columns:** enrollment\_id (Primary Key), student\_id (Foreign Key), course\_id (Foreign Key), enrollment\_date.

##### Lectures to Course Assignment

This table links lecturers to the courses they are teaching.

* **Lecturers Table Columns:** lecturer\_id (Primary Key), first\_name, last\_name, email.
* **Lectures Courses Table Columns:** lecture\_course\_id (Primary Key), lecturer\_id (Foreign Key), course\_id (Foreign Key).

##### Lectures to TA Assignment

This table records the assignment of teaching assistants to lecturers.

* **Teaching Assistants Table Columns:** ta\_id (Primary Key), first\_name, last\_name, email.
* **Lectures TA Table Columns:** lecture\_ta\_id (Primary Key), lecturer\_id (Foreign Key), ta\_id (Foreign Key).

#### 4. Data Population

Sample data was inserted into the tables to simulate a real-world scenario. This includes a few records for students, fees payments, courses, enrollments, lecturers, and teaching assistants.

#### 5. Database Function for Outstanding Fees

A PostgreSQL function was created to calculate the outstanding fees for each student. This function aggregates the total fees paid by each student and calculates the difference from the total required fees, returning the result in JSON format.

### Part 2: Next.js 14 Application Development

#### 1. Introduction

The second part of the project involved developing a web application using Next.js 14. The application includes functionalities for user login, registration, and a dashboard to display user-specific information.

#### 2. Project Setup

A new Next.js project was created using the create-next-app tool. The project structure includes pages for login, registration, and a user dashboard.

#### 4. Implementing Login, Register, and Dashboard

##### Login Page

The login page allows users to enter their credentials and authenticate themselves. On successful login, users are redirected to the dashboard.

##### Register Page

The registration page enables new users to create an account by providing their email and password. Successful registration redirects users to the login page.

##### Dashboard Page

The dashboard displays user-specific information after successful login. It fetches data from the server and presents it in a user-friendly format.

#### 5. Functionalities on the Dashboard

The dashboard is designed to provide the following useful information:

* **Welcome Message:** Personalized greeting with the user's name.
* **Student Information:** Display of personal details such as name, email, and date of birth.
* **Course Enrollments:** List of courses the student is enrolled in.
* **Fees Information:** Summary of fees paid and outstanding balance.
* **Assignments:** List of courses assigned to lecturers and teaching assistants.

### Part 3: Project Submission

#### 1. GitHub Repository

The entire project, including the Next.js application and database scripts, is hosted on GitHub. The repository includes the source code, SQL scripts for database creation and data population, and a backup of the database.

#### 2. Database Backup

A backup of the PostgreSQL database is included in the repository to facilitate easy setup and testing.

### Conclusion

The project successfully developed a relational database for the Computer Engineering Department and a Next.js application with essential functionalities. The database design ensures data integrity and efficient management, while the web application provides a user-friendly interface for interacting with the system. This project lays a solid foundation for further enhancements and integration with additional features