

**Name:** Andrews Larbi Asare  
**Student ID:** 22250041  
**Course:** CPEN 208 – Software Engineering

## Introduction

This project involves the design and development of a university student portal using the Next.js 14 framework for both frontend and backend functionalities, with PostgreSQL as the primary database. The portal supports student and lecturer registration, login, course enrollment, fee tracking, and lecturer/TA assignments.

## Technologies Used

Technology	Purpose
Next.js 14	Fullstack React Framework
PostgreSQL	Relational Database Management System
Node.js	Backend Runtime
pg & pgAdmin	PostgreSQL Client and Admin GUI
bcrypt	Password Hashing
Thunder Client	REST API Testing
Git & GitHub	Version Control and Source Hosting

## Features Implemented

- Separate dashboards for students and lecturers
- Registration and login APIs with password hashing
- Course and fee management linked to student profiles
- Lecturer and teaching assistant course assignments
- Functional PostgreSQL database schema
- Secure API communication and form validation
- Database backup and SQL migration scripts

## Authentication System

The portal uses `bcrypt` to securely hash passwords during registration. Login requests verify user credentials and return status messages accordingly.

- **Student Registration:** `/api/student/register`
- **Student Login:** `/api/student/login`
- **Lecturer Registration:** `/api/lecturer/register`
- **Lecturer Login:** `/api/lecturer/login`

Both students and lecturers have unique authentication flows and separate pages for registration and login.

## Database Design

Tables created include:

- students
- lecturers
- courses
- fees
- enrollments
- teaching\_assistants
- lecturer\_course
- lecturer\_ta

Each table uses appropriate data types and constraints. Foreign key relationships ensure referential integrity between entities such as students and their fees or course enrollments.

## API Testing

All API endpoints were tested using **Thunder Client** in Visual Studio Code. Both successful and failed request cases (e.g., invalid credentials, missing fields) were verified.

Sample test for student login:

- **Method:** POST
- **URL:** `http://localhost:3000/api/student/login`
- **Body:**

```
json
CopyEdit
{
  "email": "student@example.com",
  "password": "example123"
}
```

## Database Backup and SQL Scripts

The following scripts are included in the `sql/` directory:

- `create_tables.sql` – Creates all necessary tables
- `insert_sample_data.sql` – Inserts demo student, lecturer, and course data
- `backup.sql` – Full backup of the PostgreSQL database (exported using `pg_dump`)

## GitHub Repository

The full source code, SQL scripts, and report have been pushed to the following GitHub repository:

**Repository URL:**

[https://github.com/andrews3080/software\\_eng\\_ass](https://github.com/andrews3080/software_eng_ass)

## **Conclusion**

This project demonstrates how modern web technologies can be combined to build a fullstack academic portal. By leveraging Next.js 14 and PostgreSQL, the portal achieves a secure, scalable, and maintainable architecture suitable for a university system.

The assignment requirements were fully met, and the portal supports real-world academic operations such as user authentication, fee management, and course registration.