Name: Edmond Mawuli

ID:22031160

Course: SENG 208

Computer Engineering Department Software Portal - Project Report

1. Introduction

This project is a comprehensive software system built for the Computer Engineering Department. It includes a relational PostgreSQL database and a web application developed using Next.js 14. The system manages student information, course enrollments, fee payments, and staff (lecturer and TA) assignments. The final solution enables secure user authentication, a responsive dashboard, and administrative features for managing department activities.

2. Technologies Used

- **Frontend:** Next.js 14, React, Tailwind CSS, TypeScript
- **Backend/API:** Next.js API routes, Prisma ORM
- **Database:** PostgreSQL
- **Authentication:** bcrypt (password hashing)
- **Version Control:** Git & GitHub

3. Database Design (PostgreSQL)

3.1 Schema and Tables

A PostgreSQL database `postgres` was created. Prisma ORM was used for defining and managing the schema. The following main tables were implemented:

- `students`: Stores student personal data
- `fee_payments`: Records all student payments
- `courses`: Contains available courses
- `course_enrollments`: Links students to their enrolled courses
- `lecturers` and `teaching_assistants`: Stores staff info
- `course_lecturers`: Maps courses to lecturers
- `lecturer_tas`: Maps lecturers to TAs

3.2 Sample Data

Insert scripts were created and used to populate the database with realistic sample data using class records.

3.3 Functions

A PostgreSQL function `get_outstanding_fees()` was written to:

- Calculate the total fees paid per student
- Return outstanding balances
- Output results as a JSON array

4. Web Application (Next.js 14)

4.1 Project Setup

Created using:

```
npx create-next-app@latest compeng-portal --ts --app
```

Selected all optional tools (Tailwind, ESLint, Prisma, etc.).

4.2 Folder Structure

- `src/app/`: App Router structure with pages and API routes
- `src/components/`: Reusable React components (e.g., AuthForm, Table)
- `src/lib/prisma.ts`: Prisma client setup
- `prisma/schema.prisma`: Data models

4.3 Authentication

Implemented secure authentication with:

- Register & Login pages
- Passwords hashed using `bcrypt`
- Redirects to dashboard on success

4.4 Dashboard Features

The dashboard shows:

- Outstanding fees per student
- Enrolled courses per student
- Course assignments for lecturers and TAs

5. GitHub Repository

All source code and resources were uploaded to GitHub:

- Source files (Next.js frontend + backend)
- Prisma schema and migrations
- Database backup: `backup.sql`
- Project Report: `project-report-22031160.pdf`

6. Running the Project

Setup

1. Clone repo from GitHub
2. Run `npm install`
3. Create `.env` from `.env.copy`
4. Apply DB schema: `npx prisma db push`
5. Start dev server: `npm run dev`

7. Conclusion

This project demonstrates the full-stack integration of a relational PostgreSQL database with a modern frontend application. The system is scalable and designed for further enhancements, such as real-time notifications, email integration, and admin user roles.