

CPEN 206: Full-Stack Web Application — Student Fee & Enrollment Dashboard

- **Name:** Manal Abdul-Kadir Mohammed
 - **Index Number:** 22020624
 - **Course:** CPEN 206
 - **GitHub Repo:** <https://github.com/manal37-glitch/cpen206-Assignment>
-

1. Introduction

This project is a full-stack web application built using **Next.js 14** and **PostgreSQL**. It allows users to register, log in, and view a dashboard with student fee and course enrollment data pulled from a relational database.

2. Functionalities Implemented

- User registration and login system
 - Dashboard showing:
 - Outstanding fees for students
 - Enrollment records (course, lecturer, TA)
 - Data pulled from a PostgreSQL database via API routes
 - Responsive design with minimalistic lavender-themed UI
 - Authentication logic using localStorage (can be improved in future)
 - Fully styled login, register, and dashboard pages
-

3. Technologies Used

Layer	Tech Stack
Frontend	Next.js 14, React, TypeScript, CSS
Backend	Node.js (via Next.js App Router)

Layer	Tech Stack
Database	PostgreSQL
Styling	Global CSS with custom colors and layout
Version Control	Git + GitHub

4. Database Design

The database was created using the following schema:

- **students**
- **lecturers**
- **teaching_assistants**
- **courses**
- **enrollments**
- **payments**

A PostgreSQL function called `outstanding_fees()` was written to compute fee balances.

The SQL scripts are included in the GitHub repo under `/sql/`.

5. Pages Overview

`/register`

- Simple registration form
- Saves name, email, password to the DB

`/login`

- Authenticates user and redirects to dashboard

`/dashboard`

- Protected route (only accessible after login)
- Fetches data from:

- api/fees for fee balances
 - api/enrollments for student-course data
-

6. Deployment & Testing

This project was tested locally using:

- **PostgreSQL on localhost**
- **Next.js dev server** (npm run dev)
- GitHub was used for version control and code backup.

Due to time constraints, no deployment (e.g., Vercel) was done.

7. Database Backup

A complete database backup was created using pg_dump and is included in the repo under:

/backup/backup.sql

8. Challenges Faced

- First time using Next.js and React; had to learn components, routing, and API routes quickly
 - Encountered errors with Git, database connection, and PostgreSQL environment setup
 - Learned to debug by reading error messages and Googling stack traces
-

9. Conclusion

Despite initial unfamiliarity, this project helped me understand the flow of full-stack development: from setting up the database to building a working frontend and connecting it via API routes. I am proud of how much I learned in such a short time.
