Project Title:

Department Management System – Computer Engineering

Part 1: Database Development (PostgreSQL)

Goal:

To develop a **relational database** that supports key academic and administrative functions of the Computer Engineering department. The database is built to support the backend of a software system with the following features:

Required Functionalities:

1. **Student Personal Information**
2. **Student Fees Payments**
3. **Course Enrollment**
4. **Lecturer–Course Assignment**
5. **Lecturer–Teaching Assistant (TA) Assignment**

Database Design

Database Name:

`classcpen`

Schema:

Used default `public` schema

Tables Created:

Table Name	Description
`students`	Stores personal info like name, email, gender, phone
`fees`	Records all fees paid by each student
`courses`	Contains course titles, codes, and credit hours
`enrollments`	Tracks which student is enrolled in which course
`lecturers`	Stores lecturer details including contact info
`teaching_assistants`	Stores TA info and links them to lecturers and courses

Sample Data:

Created **INSERT scripts** to populate each table using sample data from my class:

- 63 sample students
- 10 sample lecturers
- 10 sample courses
- 10 teaching assistants
- 63 course enrollments
- Multiple fee payment records

Database Function

Function Name: `get_outstanding_fees()`
Purpose: Calculates outstanding fees per student
Returns: JSON array with student name, total amount paid, and balance

```sql
CREATE OR REPLACE FUNCTION get_outstanding_fees()
RETURNS JSON AS $$
BEGIN
  RETURN (
    SELECT json_agg(result)
    FROM (
      SELECT
        s.id,
        s.first_name || ' ' || s.last_name AS name,
        SUM(f.amount) AS amount_paid,
        s.total_fees - COALESCE(SUM(f.amount), 0) AS balance
      FROM students s
      LEFT JOIN fees f ON f.student_id = s.id
      GROUP BY s.id
    ) result
  );
END;
$$ LANGUAGE plpgsql;
```

Backup File

A full `.sql` backup of the database was included in the GitHub repository.

Part 2: Next.js 14 Application

Goal:

To develop a **full-stack application** that connects to the PostgreSQL database and provides:

- User registration

- User login and session control
- A modern dashboard showing:
    - Students
    - Lecturers
    - Courses
    - TA assignments

Tech Stack

Layer	Tech
Frontend	Next.js 14 + Tailwind CSS
Backend/API	Route Handlers (Next.js App Router)
Database	PostgreSQL
ORM	Prisma (optional depending on setup)
Auth	Cookie-based session (no 3rd-party auth)

Key Features

Authentication:

- User login and register pages
- Sessions managed via cookies
- Protected routes redirect unauthenticated users

Dashboard:

- Top navbar with logout
- Grid of square cards for **Students, Lecturers, Courses**
- On click, card expands to show detailed info
- Responsive and mobile-friendly UI

Repository Content:

- `/app` – Next.js source code
- `/db` – Database connection and queries
- `SQL_scripts with Backup` – All CREATE + INSERT + FUNCTION scripts

GitHub URL

GitHub Repository:
https://github.com/joe-qodes/next_js_project.git

Conclusion

This project successfully combines backend (PostgreSQL) and frontend (Next.js 14) to create a fully functional web-based department management system. All required features — from student fee tracking to course and TA management — have been implemented and tested with sample data. The application is scalable, modular, and can easily be extended with additional features like reporting, notifications, and user roles.