

# LAB ASSIGNMENT REPORT

## UNIVERSITY OF GHANA

Department of Computer Engineering

Project Title: Authentication App Using Next.js 14 and PostgreSQL

Name: Jessica Amemor

ID: 22066228

CPEN 208

## Introduction

This report details the development of the Next.js Student Management System, a web application built with Next.js 14. The primary goal of this project was to create a functional application with user authentication (login and registration) and a central dashboard to display key information from a database, such as student, course, and lecturer data. The core technologies used include:

- **Framework:** Next.js 14 (using the App Router)
- **Language:** TypeScript
- **Styling:** Tailwind CSS
- **Database:** PostgreSQL
- **ORM:** Prisma

## Functionalities Implemented

- **User Registration:** Stores user information (name, email, password) in database
- **User Login:** Authenticates user and creates session
- **Dashboard:** Protected route only accessible to logged-in users
- **Database Integration:** PostgreSQL handles persistent user data
- **Security:** Passwords are hashed before storage (e.g., bcrypt)

## Development and Implementation Details

The development process involved setting up the structure of the project, establishing a database connection, building a backend API to fetch data and creating a frontend interface to display it.

### 1. Backend and Database

The application's backend is powered by Next.js API Routes and a PostgreSQL database.

- **Database Schema:** The database structure was defined using Prisma, a modern Object-Relational Mapper (ORM). The schema, located in “prisma/schema.prisma”, specifies the models for students, courses, lecturers and their relationships.
- **Data Fetching:** An API endpoint was created at “/api/dashboard-data”. When the dashboard page is loaded, this endpoint is called. It uses Prisma Client to query the PostgreSQL database and retrieve the lists of all students, courses, and lecturers.

- **Database Connection:** A secure connection to the database was established by storing the database URL in an `.env` file, which is kept out of version control for security. The Prisma client was regenerated using the “`npx prisma generate`” command to ensure it was perfectly synchronized with the database schema.

## 2. Frontend and User Interface

The user interface was built using React and styled with Tailwind CSS.

- **Dashboard:** The main dashboard page at “`/dashboard`” serves as the central hub for viewing data. It contains three tables to display the information for students, courses, and lecturers fetched from the API.
- **Styling:** The entire user interface, including the sidebar navigation and data tables, was styled using the utility-first classes provided by Tailwind CSS. This allowed for rapid and consistent styling.

## 3. Version Control

The project's source code is managed with Git and is hosted on a GitHub repository.

### How the app works

1. User registers through the register page.
2. Data is sent to API route and saved to PostgreSQL.
3. On login, credentials are checked against the database.
4. Upon success, a session is started and user is redirected to the dashboard.

### Files submitted

- Source Code
- Database file
- Database Backup
- Project Report