# NAME: PRINCE HENRY AFEDI DADEBO

# ID: 22123779

# STUDENT MANAGEMENT SYSTEM

## Introduction

The Student Management System is a web-based application designed to simplify the management of student academic information, course enrollment, and fee payments. The project was developed using **Next.js 14** for the frontend, **Prisma ORM** for database communication, and **PostgreSQL** as the database. It provides both students and administrators with an efficient way to track academic progress, outstanding fees, and other essential student information.

The system focuses on essential features such as **student registration**, **login authentication**, **dashboard overview**, **course management**, and **fee summaries**.

## Project Planning and Design

The project started with identifying key functionalities needed for effective student management. These include:

* User Authentication (Registration & Login)
* Dashboard to view student information
* Course enrollment view
* Fees summary and breakdown

Once these core features were established, the design followed modern **UI/UX principles**, using **Tailwind CSS** and **shadcn/ui components** to ensure a clean and responsive interface.

The application follows a **REST API architecture**, separating frontend display logic from backend data management. For database interactions, Prisma ORM was used to map JavaScript objects to relational database tables efficiently.

## Technologies Used

**Frontend:**

* Next.js 14 (App Router)
* React.js
* Tailwind CSS
* shadcn/ui
* Lucide-react icons

**Backend/API:**

* Next.js API Routes
* Prisma ORM
* bcryptjs (for password hashing)
* JSON Web Tokens (JWT) for authentication.

**Database:**

* PostgreSQL

## Application Features

### User Registration and Login

The registration feature collects student details such as name, email, password, year of study, and optional personal information like phone number and address. Passwords are securely hashed using **bcryptjs** before being stored in the database.

The login system uses **JWT tokens** for user authentication, providing secure access to protected dashboard routes.

### Student Dashboard

Once logged in, users access a **personalized dashboard**. The dashboard displays:

* Year of study and enrollment status
* Total number of enrolled courses
* Outstanding balance and total payments
* Recent academic activities (e.g., payments, enrollments)
* Fee breakdown for the academic year

### Courses Overview

Students can view a list of their current enrollments with details such as:

* Course code and name
* Assigned lecturer
* Enrollment status

### Fees Summary

The fee management section shows:

* Total amount due
* Total payments made
* Outstanding balance

This helps students track their financial obligations and deadlines easily.

## Challenges Encountered

One of the main challenges during development was **managing form validation** for registration and login while maintaining clean API responses. Another challenge involved ensuring that fee breakdowns were accurately calculated and displayed, especially when handling mock data before database integration.

Also, building a fully responsive interface required careful adjustments of component layouts using Tailwind’s grid and flex utilities.

## Conclusion

The Student Management System successfully demonstrates the integration of modern web technologies to solve a common academic need. By using Next.js, React, PostgreSQL, and Prisma, the project provides a solid foundation for future development, including features like admin panels, payment integration, and grading systems.

The application was built with scalability in mind, allowing additional modules such as student transcripts, detailed grade reports, and real payment integrations to be implemented in future versions.

## References

## Next.js Documentation: <https://nextjs.org/docs>

## React Documentation: <https://react.dev/learn>

## Prisma ORM Documentation: <https://www.prisma.io/docs>

## PostgreSQL Official Documentation: [https://www.postgresql.org/docs/](https://www.postgresql.org/docs/" \t "_new)

## Tailwind CSS Documentation: [https://tailwindcss.com/docs](https://tailwindcss.com/docs" \t "_new)

## bcryptjs GitHub Repository: [https://github.com/dcodeIO/bcrypt.js](https://github.com/dcodeIO/bcrypt.js" \t "_new)

## Lucide React Icons Documentation: [https://lucide.dev/](https://lucide.dev/" \t "_new)

## Mozilla Developer Network (MDN) Web Docs: [https://developer.mozilla.org/en-US/](https://developer.mozilla.org/en-US/" \t "_new)

## JavaScript Info Tutorials: [https://javascript.info/](https://javascript.info/" \t "_new)

## Shadcn UI Components Documentation: https://ui.shadcn.com/