11348310

Report on the Creation of Next.js-based Login, Register, and Dashboard Pages using TypeScript and PostgreSQL

Introduction

This report details the creation of a student management system using Next.js, TypeScript, and PostgreSQL. The system includes a login page, a registration page, and a dashboard, which are essential for user management. The database schema has been designed to manage students, courses, lecturers, and other related entities efficiently.

Database Design

The database schema for the student management system is defined using PostgreSQL. Below is an overview of the tables and their relationships:

1. Students Table

The `students` table stores personal information about each student.

- **Columns**:
  - `student_id`: Unique identifier for each student (Primary Key).
  - `last_name`: The last name of the student.
  - `first_name`: The first name of the student.
  - `email`: Unique email address of the student.
  - `date_of_birth`: The date of birth of the student.

2. Fees Table

The `fees` table records the fees paid by each student.

- **Columns**:
  - `fee_id`: Unique identifier for each fee transaction (Primary Key).
  - `student_id`: Identifier of the student who made the payment (Foreign Key referencing `students(student_id)`).
  - `amount`: The amount paid.
  - `date_paid`: The date the payment was made.

3. Courses Table

The `courses` table stores information about different courses.

- **Columns**:
  - `course_code`: Unique code for each course (Primary Key).

  - course_name: Name of the course.
  - course_description: Description of the course.

4. Lecturers Table

The lecturers table stores information about the lecturers.

- **Columns**:
  - lecturer_id: Unique identifier for each lecturer (Primary Key).
  - last_name: The last name of the lecturer.
  - first_name: The first name of the lecturer.
  - email: Unique email address of the lecturer.

5. Course Enrollments Table

The course_enrollments table records which students are enrolled in which courses.

- **Columns**:
  - enrollment_id: Unique identifier for each enrollment (Primary Key).
  - student_id: Identifier of the student (Foreign Key referencing students(student_id)).
  - course_code: Code of the course (Foreign Key referencing courses(course_code)).
  - enrollment_date: The date the student enrolled in the course.

6. Lecturer Course Assignments Table

The lecturer_course_assignments table records which lecturers are assigned to which courses.

- **Columns**:
  - assignment_id: Unique identifier for each assignment (Primary Key).
  - lecturer_id: Identifier of the lecturer (Foreign Key referencing lecturers(lecturer_id)).
  - course_code: Code of the course (Foreign Key referencing courses(course_code)).
  - assignment_date: The date the lecturer was assigned to the course.

7. Lecturer TA Assignments Table

The lecturer_ta_assignments table records which teaching assistants are assigned to which lecturers.

- **Columns**:
  - assignment_id: Unique identifier for each assignment (Primary Key).
  - lecturer_id: Identifier of the lecturer (Foreign Key referencing lecturers(lecturer_id)).
  - ta_id: Identifier of the teaching assistant (Foreign Key referencing teaching_assistants(ta_id)).
  - assignment_date: The date the teaching assistant was assigned to the lecturer.

8. Teaching Assistants Table

The teaching_assistants table stores information about the teaching assistants.

- **Columns**:
  - ta_id: Unique identifier for each teaching assistant (Primary Key).
  - last_name: The last name of the teaching assistant.
  - first_name: The first name of the teaching assistant.
  - email: Unique email address of the teaching assistant.

Creation of Next.js Pages

1. Register Page

The *RegisterPage* component allows users to create a new account. It includes fields for entering a student or staff ID, a password, and a confirmation password.

2. Login Page

The *LoginPage* component enables users to log into the system using their credentials. It includes fields for entering an ID and a password.

3. Dashboard Page

The *DashboardPage* component serves as the main interface for users once they have logged in. It provides access to various features of the student management system, such as viewing courses, managing personal information, and accessing fee payment history.

Conclusion

This report outlines the essential components and design considerations for creating a student management system using Next.js, TypeScript, and PostgreSQL. The provided database schema supports efficient data management for students, courses, lecturers, and related entities, while the Next.js pages offer a user-friendly interface for interacting with the system.