

SHỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



BÁO CÁO THỰC TẬP CƠ SỞ TUẦN 9

Bắt đầu thực hiện backend

Giảng viên hướng dẫn	: TS. Kim Ngọc Bách
Họ và tên sinh viên	: Phạm Trung Kiên
Mã sinh viên	: B22DCVT263
Lớp	: E22CQCN02-B

Hà Nội – 2025

I. Mục tiêu của tuần

- Thiết kế mô hình dữ liệu cho hệ thống quản lý khóa học, điểm số và người dùng.
- Kết nối và đồng bộ cơ sở dữ liệu với PostgreSQL bằng Prisma ORM.
- Phát triển một số API cơ bản phục vụ cho việc đăng ký, tính toán GPA và quản lý dữ liệu người dùng – khóa học.

II. Nội dung công việc chi tiết

1. Thiết kế cơ sở dữ liệu PostgreSQL

a. Các bảng chính

- **User**: Lưu thông tin cá nhân của người dùng (sinh viên).
- **Course**: Thông tin về các môn học (tên, tín chỉ, học kỳ).
- **Grade**: Kết nối giữa sinh viên và môn học, lưu điểm số.
- **Semester**: Tên học kỳ và năm học.

b. Mối quan hệ giữa các bảng

- Một **User** có thể có nhiều **Grade**.
- Mỗi **Grade** gắn với một **Course** cụ thể.
- Một **Course** thuộc về một **Semester** nhất định.

c. Ví dụ định nghĩa trong Prisma schema:

prisma

```
model User {  
  id      Int      @id @default(autoincrement())  
  fullName String  
  email    String   @unique  
  password String  
  createdAt DateTime @default(now())  
  grades   Grade[]  
}
```

```
model Course {  
  id      Int    @id @default(autoincrement())  
  title   String  
  credit  Int  
  semester Semester @relation(fields: [semesterId], references: [id])  
  semesterId Int  
  grades  Grade[]  
}
```

```
model Grade {  
  id      Int    @id @default(autoincrement())  
  score   Float  
  user    User   @relation(fields: [userId], references: [id])  
  userId  Int  
  course  Course @relation(fields: [courseId], references: [id])  
  courseId Int  
}
```

```
model Semester {  
  id      Int    @id @default(autoincrement())  
  name    String  
  year    Int  
  courses Course[]  
}
```

d. Đồng bộ schema với database

Migrate schema lần đầu:

```
npx prisma migrate dev --name init
```

2. Xây dựng các API cơ bản

a. Nhóm API người dùng

- `POST /api/auth/register` – Đăng ký người dùng mới.
- `POST /api/auth/login` – Đăng nhập và trả token.
- `GET /api/users/:id` – Lấy thông tin cá nhân.

b. Nhóm API khóa học

- `POST /api/courses` – Thêm môn học.
- `GET /api/courses` – Lấy toàn bộ danh sách khóa học.
- `GET /api/courses/:id` – Lấy chi tiết 1 môn học cụ thể.

c. Nhóm API điểm số

- `POST /api/grades` – Nhập điểm cho sinh viên.
- `GET /api/grades/user/:userId` – Lấy danh sách điểm theo sinh viên.
- `GET /api/gpa/:userId` – Tính GPA trung bình theo tín chỉ.

d. Ví dụ tính GPA:

javascript

```
const grades = await prisma.grade.findMany({  
  where: { userId: parseInt(userId) },  
  include: { course: true }  
});  
  
const totalCredits = grades.reduce((sum, g) => sum + g.course.credit, 0);  
const totalPoints = grades.reduce((sum, g) => sum + g.score * g.course.credit, 0);  
  
const gpa = totalPoints / totalCredits;
```

III. Kết quả đạt được

Công việc	Trạng thái
Thiết kế mô hình dữ liệu	Hoàn thành
Migrate schema với PostgreSQL	Hoàn thành
Viết API người dùng và khóa học	Hoàn thành
API tính điểm GPA hoạt động ổn định	Hoàn thành
Test thử API bằng Postman	Hoàn thành

IV. Định hướng tuần tiếp theo

- Hoàn thiện các API cập nhật và xóa dữ liệu.
- Tích hợp frontend hoặc API gọi từ client.
- Thêm xác thực người dùng bằng JWT.
- Viết middleware bảo vệ route.
- Viết các unit test cho endpoint chính.
- Phát triển tính năng nâng cao: thống kê, biểu đồ, xuất bảng điểm.