

SHỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



BÁO CÁO THỰC TẬP CƠ SỞ TUẦN 10

Tiếp tục phát triển 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 MongoDB
- Lý do chuyển sang MongoDB:
 - + Linh hoạt về cấu trúc dữ liệu.
 - + Hiệu quả với dữ liệu dạng JSON hoặc tài liệu lồng nhau.
 - + Tốc độ phát triển và thay đổi nhanh trong dự án
 - + Thích hợp với ứng dụng hiện đại

II. Nội dung công việc chi tiết

1. TỔNG QUAN VỀ CÔNG VIỆC THỰC HIỆN

Trong tuần vừa qua, em đã tập trung vào việc xây dựng toàn bộ phần backend cho hệ thống quản lý điểm trung bình học tập (GPA) kết hợp với chức năng tổ chức và chấm điểm các bài kiểm tra thực hành cho học sinh/sinh viên. Hệ thống được triển khai theo mô hình RESTful API, sử dụng Node.js với framework Express.js và cơ sở dữ liệu MongoDB.

Các nội dung chính em đã hoàn thành bao gồm:

- Xây dựng hệ thống xác thực người dùng sử dụng JWT (JSON Web Token)
- Tạo các API phục vụ chức năng đăng ký, đăng nhập, cập nhật thông tin người dùng
- Phát triển module tính GPA và quản lý điểm học tập
- Phát triển hệ thống quản lý bài kiểm tra thực hành, cho phép giáo viên tạo đề, học sinh làm bài và nộp kết quả
- Thiết kế cơ sở dữ liệu MongoDB với các schema phù hợp
- Kiểm thử API bằng Postman, xử lý lỗi và đảm bảo bảo mật dữ liệu

2. CHI TIẾT CÁC MODULE VÀ API ĐÃ TRIỂN KHAI

2.1. Hệ thống xác thực và phân quyền người dùng

- **Công việc thực hiện:**
 - Tạo cơ chế đăng ký tài khoản cho 3 loại người dùng: student, teacher, admin
 - Cấu hình JWT để cấp access token và refresh token
 - Tạo middleware kiểm tra token để bảo vệ các route riêng tư
- **Các API đã xây dựng:**
 - POST /api/auth/register: Đăng ký tài khoản người dùng mới
 - POST /api/auth/login: Đăng nhập, trả về access và refresh token

- POST /api/auth/logout: Đăng xuất, xóa token khỏi hệ thống
- POST /api/auth/refresh-token: Làm mới token để duy trì phiên đăng nhập
- GET /api/auth/profile: Truy xuất thông tin người dùng đã đăng nhập
- PUT /api/auth/profile: Cập nhật hồ sơ cá nhân
- PUT /api/auth/change-password: Đổi mật khẩu
- GET /api/auth/users: (Chỉ admin/giáo viên) lấy danh sách người dùng
- GET|PUT|DELETE /api/auth/users/:id: Quản lý thông tin người dùng theo ID
- **Đánh giá:** Hệ thống xác thực ổn định, phân quyền rõ ràng. Có thể mở rộng cho hệ thống phân cấp quyền truy cập phức tạp hơn.

2.2. Module GPA – Quản lý điểm trung bình học kỳ

- **Công việc thực hiện:**
 - Thiết kế schema GPA gồm subjectId, score, semester
 - Tính toán GPA trung bình theo học kỳ
 - Triển khai biểu đồ hiệu suất GPA theo thời gian
- **Các API đã xây dựng:**
 - POST /api/gpa: Thêm điểm mới cho môn học
 - GET /api/gpa: Lấy toàn bộ điểm của người dùng
 - GET /api/gpa/performance: Phân tích điểm GPA theo thời gian (dùng để vẽ biểu đồ)
 - PUT /api/gpa/:id: Cập nhật điểm
 - DELETE /api/gpa/:id: Xóa điểm khỏi hệ thống
 - POST /api/gpa/bulk: Thêm nhiều mục GPA một lần (tối ưu hiệu năng nhập liệu)
- **Đánh giá:** Dữ liệu GPA xử lý nhanh, chính xác, hỗ trợ việc học tập và theo dõi kết quả cá nhân hiệu quả.

2.3. Module Subject – Quản lý môn học

- **Công việc thực hiện:**
 - Thiết kế schema môn học đơn giản, lưu tên và mã môn
 - Cho phép tạo/xóa môn học tùy ý người dùng (hoặc giáo viên)
- **Các API đã xây dựng:**
 - GET /api/subjects: Lấy danh sách môn học
 - POST /api/subjects: Tạo môn học mới
 - DELETE /api/subjects/:value: Xóa môn học theo mã
- **Đánh giá:** Hệ thống linh hoạt, hỗ trợ việc thêm các môn học mới nhanh chóng mà không cần can thiệp vào database thủ công.

2.4. Module Practice Test – Bài kiểm tra thực hành

- **Công việc thực hiện:**
 - Cho phép giáo viên tạo đề kiểm tra với câu hỏi, đáp án
 - Học sinh xem đề và gửi kết quả làm bài
 - Hệ thống chấm điểm tự động và lưu kết quả
- **Các API đã xây dựng:**
 - POST /api/practice-tests: Tạo bài kiểm tra mới (chỉ giáo viên)
 - GET /api/practice-tests: Lấy danh sách các bài kiểm tra
 - GET /api/practice-tests/:id: Xem chi tiết bài kiểm tra
 - PUT /api/practice-tests/:id: Cập nhật bài kiểm tra
 - DELETE /api/practice-tests/:id: Xóa bài kiểm tra
 - POST /api/practice-tests/submit-result: Học sinh nộp bài làm
 - GET /api/practice-tests/dashboard-stats: Lấy thống kê số lượng làm bài, kết quả
- **Đánh giá:** Giao diện backend hỗ trợ tốt giáo viên trong việc tạo bài và theo dõi kết quả. Dễ dàng mở rộng thêm phân tích nâng cao hoặc báo cáo kết quả.

3. CƠ SỞ DỮ LIỆU MONGODB

- **Schema đã tạo:**
 - User: gồm username, email, password (hash), role
 - GPA: lưu điểm số, học kỳ, môn học, người dùng
 - Subject: tên và mã môn học
 - PracticeTest: đề kiểm tra, danh sách câu hỏi, kết quả
- **Chiến lược kết nối:**
 - Sử dụng Mongoose để định nghĩa mô hình và thao tác dữ liệu
 - Dùng populate() để kết nối các bảng liên quan
- **Đánh giá:** MongoDB hoạt động ổn định, phù hợp cho dữ liệu lồng nhau như kết quả bài làm hoặc hồ sơ người dùng.

4. KIỂM THỬ, VALIDATION VÀ BẢO MẬT

- Dùng Postman để kiểm tra toàn bộ endpoint: phương thức, đầu vào, kết quả trả về
- Áp dụng middleware validate schema bằng thư viện như Joi hoặc thủ công
- Middleware xác thực đảm bảo chỉ người dùng hợp lệ mới truy cập API nhạy cảm
- Đã xử lý các lỗi phổ biến như thiếu trường, token không hợp lệ, sai định dạng JSON

5. ĐỊNH HƯỚNG PHÁT TRIỂN TUẦN TỚI

- Tối ưu hiệu suất truy vấn bằng cách sử dụng index
- Phát triển thêm dashboard cho admin và teacher
- Bắt đầu kết nối với frontend React.js, thử nghiệm liên kết frontend-backend