BÁO CÁO DẬN - TUẦN 6

Họ và Tên: Lý Chí Công

MSV: B22DCCN088

Bộ môn: Thực tập cơ sở

1. Thông tin dự án:

Tên dự án: Hệ thống quản lý sân bóng
Công nghệ sử dụng: ReactJS (frontend), Node.js + Express.js (backend), MongoDB (database)
Mục tiêu chung: Xây dựng hệ thống quản lý sân bóng, bao gồm đặt sân, thanh toán, quản lý lịch sử sử dụng.

---

2. Mục tiêu tuần 7:

- Học và thực hành các thao tác CRUD (Create, Read, Update, Delete) trong Node.js và MongoDB.
- Viết một vài API cơ bản sử dụng các thao tác CRUD.
- Thiết lập kết nối với MongoDB (sử dụng MongoDB Driver).
- Kiểm thử các API đã viết bằng Postman.
- Xác định các collection và schema cơ bản.

3. Kết quả đạt được:

Học và thực hành CRUD:

- Đã tìm hiểu các phương thức của MongoDB Driver để thực hiện CRUD (e.g., `insertOne`, `find`, `updateOne`, `deleteOne`).
- Đã viết các hàm Node.js để thực hiện các thao tác CRUD trên MongoDB.

```
const { MongoClient } = require("mongodb")

// Biến lưu trữ kết nối database
let db = null

// Hàm kết nối đến MongoDB
const connectDB = async () => {
  try {
    if (db) return db

    const client = new MongoClient(process.env.MONGODB_URI, {
      useNewUrlParser: true,
      useUnifiedTopology: true,
    })

    await client.connect()
    console.log(`MongoDB Connected: ${client.db().databaseName}`)

    db = client.db()
    return db
  } catch (error) {
    console.error(`Error: ${error.message}`)
    process.exit(1)
  }
}

// Hàm lấy kết nối database
const getDB = () => {
  if (!db) {
    throw new Error("Database not initialized. Call connectDB first.")
```

Viết các API cơ bản:

Đã viết các API cho chức năng quản lý người dùng:

```
const { validationResult } = require("express-validator")
const asyncHandler = require("../middlewares/asyncHandler")
const ErrorResponse = require("../utils/errorResponse")
const userService = require("../services/userService")

// @desc    Get all users
// @route   GET /api/v1/users
// @access  Private/Admin
exports.getUsers = asyncHandler(async (req, res, next) => {
  const users = await userService.getAllUsers()

  res.status(200).json({
    success: true,
    count: users.length,
    data: users,
  })
})

// @desc    Get single user
// @route   GET /api/v1/users/:id
// @access  Private/Admin
exports.getUser = asyncHandler(async (req, res, next) => {
  const user = await userService.getUserById(req.params.id)

  res.status(200).json({
    success: true,
    data: user,
  })
})
```

Thiết lập kết nối MongoDB:

Đã thiết lập kết nối thành công với MongoDB (MongoDB Atlas hoặc local) bằng MongoDB Driver.

Đã cấu hình thông tin kết nối trong file `.env`.

Kiểm thử API bằng Postman:

Đã sử dụng Postman để kiểm thử các API đã viết.

Đã kiểm tra các trường hợp thành công và một số trường hợp lỗi cơ bản (e.g., trùng email, không tìm thấy người dùng).

Xác định collection và schema:

Đã xác định các collection cơ bản:

`users`: Để lưu thông tin người dùng.

`fields`: Để lưu thông tin sân bóng.

Đã xác định schema đơn giản cho các collection (ví dụ: các trường của `users` là `email`, `password`, `name`).

4. Khó khăn và hướng giải quyết:

Khó khăn:

Xử lý các trường hợp lỗi một cách chi tiết và trả về mã trạng thái HTTP phù hợp.

Viết các truy vấn MongoDB hiệu quả (ví dụ: sử dụng index).

Hiểu rõ về cách hoạt động của các middleware trong Express.js.

Hướng giải quyết:

Tham khảo tài liệu về HTTP status codes và best practices cho REST API.

Đọc tài liệu MongoDB về indexing và optimization.

Thực hành viết các middleware đơn giản và nâng cao trong Express.js.

5. Mục tiêu tuần 8:

Hoàn thiện các API quản lý người dùng (thêm validation, authentication).

Viết các API CRUD cho chức năng quản lý sân bóng.

Tìm hiểu về authentication (xác thực) và authorization (phân quyền).

Bắt đầu cấu hình CORS.