LÂM TỀ HÀO - 1612175

**Môn học:** Phát triển ứng dụng web nâng cao CQ2016/31

**GVHD:** NGUYỄN HUY KHÁNH

MÔ TẢ CHỨC NĂNG

# Tạo project bằng express-generator và kết nối với CSDL:

Tạo project bằng express-generator, cài đặt các thư viện passpostjs, jwt, sequelize, mysql2,…

Tổ chức các thư mục chính:

* Root: chứa passport.js và db.js (đồng bộ với CSDL mysql local), app.js.
* Routes: chứa auth.js (route dùng để chạy API POST đăng nhập), users.js (route dùng để chạy API GET lấy thông tin người dùng), index.js (route dùng để lấy thông tin tất cả người dùng đã đăng ký và chạy API GET đăng ký tài khoản mới).
* Model: user.js (chứa thông tin tạo bảng người dùng trong cơ sở dữ liệu, bảng sẽ được tạo theo code nếu như chưa tồn tại bảng và là model của đối tượng trả về từ CSDL).

Kết nối với CSDL mysql bằng thư viện sequelize.

# Commit và Push lên Git theo giai đoạn:

Các lần commit & push:

* Create express app: tạo project bằng express-generator.
* Install passport-jwt, jsonwebtoken: cài đặt các thư viện của passportjs và jwt vào project.
* Create routes: cài đặt các routes và cài đặt chức năng login và passport.js.
* Create user profile return: cài đặt hàm trả thông tin user khi đăng nhập thành công.
* Connect to mysql database: kết nối với csdl mysql.
* Modify database connection: kết nối lại với csdl mysql bằng thư viện sequelize.
* Reconnect with database via sequelize: cài đặt chức năng đăng ký/ tạo người dùng lên csdl.
* Return user’s info (email only) when logged in sucessfully: cài đặt lại các routes, chỉnh sửa các hàm xác thực qua passportjs, cài đặt lại chức năng trả về thông tin người dùng (chỉ trả về email đăng ký) và token sau khi đăng nhập thành công.

Đường dẫn đến repository trên GitHub: <https://github.com/vincentlam3011/RestfulAPI>

# Tạo API POST /register để tạo người dùng:

Đăng ký người dùng mới qua api /user/register với tham số là body json gồm 2 thuộc tính “email” và “password” (lưu ý chưa cài đặt hash password). Tạo thành công và cập nhật vào CSDL.

# Tạo API POST /login để đăng nhập

Dùng passport-local để kiểm tra người dùng đăng nhập đúng hay không. Nếu không thành công thì báo lỗi tương ứng (không nhập email/ password hoặc password không đúng hoặc tài khoản không tồn tại). Nếu thành công, trả về token bí mật mà JWT trả ra và token này được dùng để xác thực.

# Tạo API GET /me để lấy thông tin người dùng hiện tại sau khi đăng nhập thành công:

Sau khi đăng nhập thành công và nhận được chuỗi token JWT trả ra, tiến hành copy token này và dùng tham số truyền vào từ url (UrlQueryParam) với biến tham số có tên là ‘secret\_token’ (key), và giá trị (value) là chuỗi token ta nhận được. Nhập đủ tên tham số và giá trị (nếu đúng) sẽ trả ra thông tin người dùng kèm thông báo đã được cấp quyền và ngược lại, nhận được thông báo chưa được cấp quyền.

Tất cả các chức năng đều được test bằng postman. Với 2 chức năng register và login nhận tham số là body json và lấy thông tin sau khi đăng nhập là tham số truy vấn trên url.

# Upload lên Host:

*\*Chưa hoàn thành\**