

LẬP TRÌNH WEB NÂNG CAO - 503106

BÀI TẬP THỰC HÀNH TUẦN 2

Giảng viên biên soạn: ThS. Mai Văn Mạnh

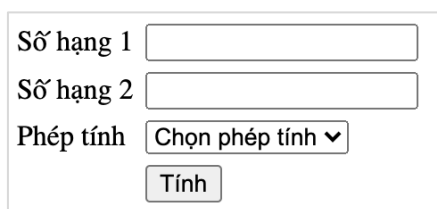
MỤC TIÊU BÀI THỰC HÀNH:

1. Cài đặt NodeJS và làm quen với công cụ quản lý module [npm](#).
2. Tạo web server và ứng dụng web đơn giản với NodeJS.
3. Tìm hiểu các khái niệm cơ bản trong NodeJS như: [REPL](#), [callback](#), [event emitter](#) và [stream](#), [modules](#) ...
4. Cách xử lý [request](#) và [response](#) trong NodeJS.
5. Xử lý tập tin, xử lý URL và query string bằng các module được tích hợp sẵn như [fs](#), [url](#), [querystring](#) và [path](#).
6. Cũng cố lại các kiến thức liên quan đến HTTP Protocol như [headers](#), [status code](#), [HTTP methods](#)...
7. Xử lý HTML form đơn giản.

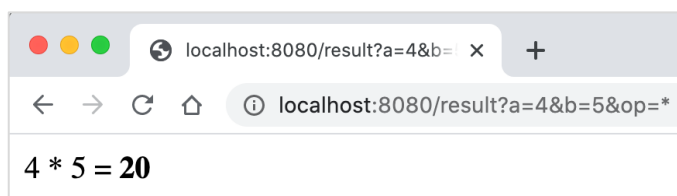
Các bài tập dưới đây yêu cầu chỉ sử dụng các module có sẵn trong NodeJS, không sử dụng thêm bất kỳ module nào khác được cài từ package manager (npm, yarn).

Bài 1. Bài tập tạo trang web cơ bản bằng [http](#) module trong NodeJS.

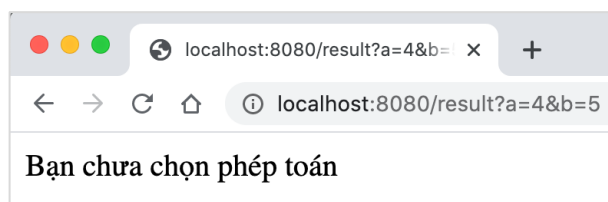
Sử dụng [http](#) module để tạo một trang web tính toán cơ bản như hình bên dưới. Khi truy cập <http://localhost/> thì trang web sẽ hiển thị một html form như hình bên dưới. Khi nhấn nút “**Tính**”, kết quả sẽ được hiển thị tại <http://localhost/result>.



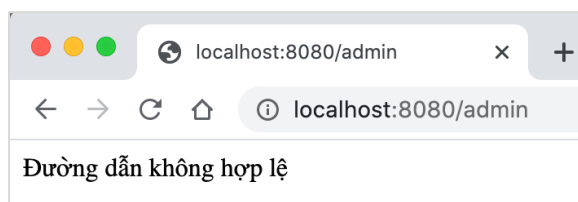
Hình 1. Giao diện trang chủ hiển thị form nhập số



Hình 2. Giao diện hiển thị kết quả khi nhập đủ thông tin



Hình 3. Giao diện hiển thị lỗi khi nhập thiếu thông tin



Hình 4. Thông báo lỗi khi truy cập một url không hỗ trợ

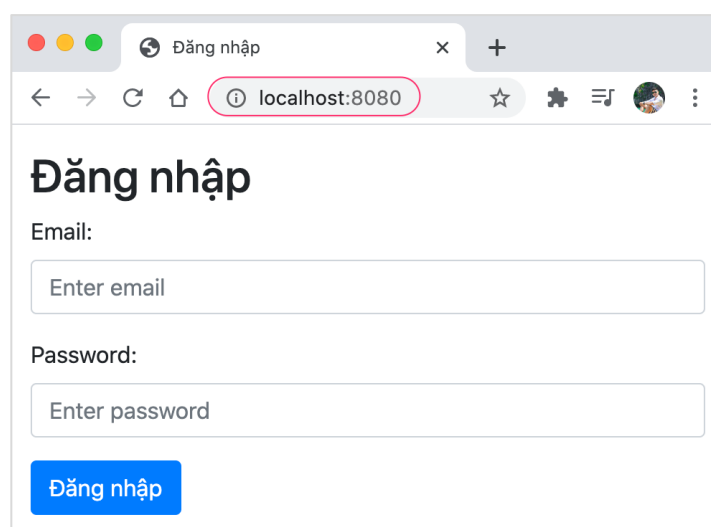
Yêu cầu:

- Toàn bộ chương trình chỉ được viết trong **một tập tin main.js** duy nhất.
- Kiểm tra lỗi nếu có và hiển thị thông báo lỗi phù hợp.

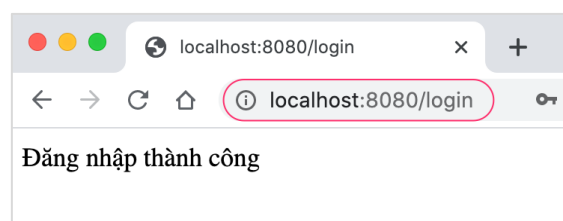
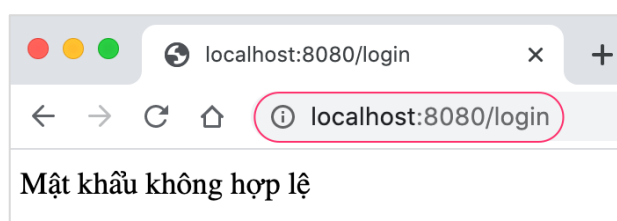
Bài 2. Bài tập xử lý POST request bằng **http** module

Thực hiện một bài tập tương tự như bài tập 1.

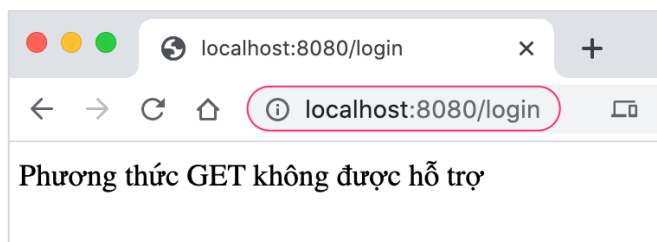
- <http://localhost>: hiển thị một form đăng nhập, form này tiếp nhận hai thông tin là email và mật khẩu rồi chuyển đến </login> bằng phương thức POST.
- <http://localhost/login>: tiếp nhận thông tin đăng nhập được gửi đến qua phương thức POST và xử lý việc đăng nhập, hiển thị thông báo lỗi nếu có. Khi đăng nhập thành công thì hiển thị thông báo '*đăng nhập thành công*'.
- Yêu cầu khác:
 - Nội dung của các trang web cần được lấy từ các tập tin html thông qua cơ chế đọc file của **fs** module.



Giao diện trang chủ hiển thị form đăng nhập

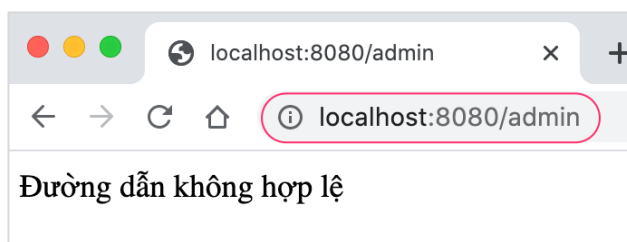


Thông báo lỗi khi không nhập mật khẩu



Thông báo lỗi khi truy cập /login bằng GET

Thông báo đăng nhập thành công



Thông báo lỗi khi truy cập các url khác

Bài 3. Sử dụng NodeJS để tạo một Rest API cung cấp thông tin cơ bản về sinh viên như sau.

- <http://localhost/students>:
 - o Phương thức GET: Trả về danh sách sinh viên dạng JSON Array.
 - o Phương thức POST: Thêm một sinh viên mới
- <http://localhost/students/{id}>:
 - o Phương thức GET: Trả về thông tin của một sinh viên cụ thể
 - o Phương thức PUT: Cập nhật thông tin mới cho sinh viên
 - o Phương thức DELETE: Xóa sinh viên
- Truy cập các endpoints khác: trả về thông báo lỗi dạng JSON.

Lưu ý:

- Dữ liệu mẫu do sinh viên tự khởi tạo.
- Chỉ cần tạo API, sau đó sử dụng các công cụ REST client để kiểm tra API.