

CSC13010 - Thiết kế phần mềm

Multiple Languages Solution

Mục lục

I. Thông tin thành viên	1
II. Solution	1
A. Front End	1
B. Back End	2

I. Thông tin thành viên

Họ tên	MSSV
Giang Đức Nhật	22120252
Nguyễn Quốc Tường	22120413
Nguyễn Trường Vũ	22120441
Bùi Đình Gia Vỹ	22120450

II. Solution

A. Front End

- Về phía Front end, ứng dụng sẽ sử dụng thư viện i18n kết hợp với 2 file json (en-translation và vn-translation) để mapping giúp hiển thị các thành phần trên website với hai ngôn ngữ Tiếng Việt và Tiếng Anh.
- Việc sử dụng thư viện i18n thay vì xử lý thủ công đem lại khá nhiều lợi ích cho ứng dụng hiện tại và cả sau này:
 - Về dễ quản lý, bảo trì, thay vì viết các hàm logic để xử lý thủ công, tạo các Object lưu trữ lớn rồi truyền thông qua các Props hoặc Context, i18n cho phép người dùng tập trung các key dữ liệu trong JSON và chỉ cần gọi t('key'), thông qua đó có thể dễ dàng scale cho dự án mới.
 - Về xử lý ngoại lệ, i18n có hỗ trợ LanguageDetector giúp phát hiện các ngôn ngữ của hệ thống, cookie, local storage, ... và sẽ fallback nếu như ngôn ngữ sử dụng không hỗ trợ

- Về Interpolation và Pluralization, i18n có thể hỗ trợ dev trong việc truyền biến vào key, thông qua đó dễ dàng tùy biến (VD khi muốn hiển thị “Hello, Vy” hoặc “Hello, John” thì i18n hỗ trợ truyền thông qua t(‘Hello’, { name: ‘John’ })))
- Về hiệu suất, i18n hỗ trợ lazy loading và dynamic import giúp tối ưu hóa hiệu suất

B. Back End

- Vì ứng dụng cho phép người dùng nhập text thoải mái nên họ sẽ có thể nhập ngôn ngữ của mình. Chính vì mọi dữ liệu được server xử lý là dữ liệu động và được lưu theo ngôn ngữ mà người dùng sử dụng để nhập vào. Do đó phía backend không cần xử lý đa ngôn ngữ.
- Trong trường hợp cần xử lý đa ngôn ngữ, chẳng hạn như nếu dữ liệu ngành học, môn học là cố định trong ứng dụng thì ta có thể tạo một bộ mapping bằng json và map dựa vào path variable trên url (ví dụ: <http://localhost:8080/en/students>) ta sẽ dựa vào locale mà chọn bộ mapping phù hợp (trong ví dụ này locale là en tức tiếng Anh).