

ĐẠI HỌC QUỐC GIA TP HCM
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO PHƯƠNG ÁN HỖ TRỢ ĐA NGÔN NGỮ

Nhóm thực hiện: 17 - niceScore
Thành viên: 22120062 Nguyễn Đăng Điền
22120186 Huỳnh Tấn Lộc
22120199 Trần Lượng
Lớp: Thiết kế phần mềm (22_3)

Tp HCM – Tháng 5/2025

Mục lục

I.	Phương án hỗ trợ đa ngôn ngữ cho các nội dung tĩnh	1
1.	Mô tả phương án	1
2.	Ưu điểm và nhược điểm của phương án	1
a)	Ưu điểm	1
b)	Nhược điểm.....	1
3.	Các phương án dự phòng khi hệ thống lớn.....	2
a)	Chia file json thành nhiều file nhỏ dựa theo views.....	2
b)	Dùng thư viện hỗ trợ i18n chuyên sâu	2
c)	Kết nối với nền tảng quản lý dịch	2
II.	Phương án hỗ trợ đa ngôn ngữ cho các nội dung động.....	3
1.	Mô tả phương án	3
2.	Các bảng và nội dung cần thay đổi	4
a)	Bảng Module.....	4
b)	Bảng Class	4
c)	Bảng Faculty – Khoa	4
d)	Bảng Status – Trạng thái	4
e)	Thông tin sinh viên (student)	4
3.	Ưu điểm và nhược điểm của phương án	5
a)	Ưu điểm	5
b)	Nhược điểm.....	5
4.	Các phương án dự phòng khi hệ thống lớn.....	5
a)	Sử dụng dịch vụ đa ngôn ngữ bên ngoài.....	5
III.	Tài liệu tham khảo.....	6

I. Phương án hỗ trợ đa ngôn ngữ cho các nội dung tĩnh

1. Mô tả phương án

- Hệ thống sử dụng cơ chế ánh xạ key-value thông qua các tệp JSON (*en.json*, *vi.json*) để hỗ trợ đa ngôn ngữ. Các khóa dịch (translation key) được sử dụng trong giao diện bằng cú pháp `{{t "table.student"}}`, với *t* là helper được cấu hình trong *express-handlebars*. Các file *en.json*, *vi.json* được lưu trong thư mục *locales*.

//en.json

```
{
  "index": {
    "student": "Student"
  }
}
```

//vi.json

```
{
  "index": {
    "student": "Sinh viên"
  }
}
```

- Ngôn ngữ được chọn thông qua *req.query.lang*, mặc định là *'en'* nếu không chỉ định.
- Truyền file json vào file JS thông qua *load file*, với ngôn ngữ từ *lang*.
- Có nút chọn thả xuống (*dropdown button*) chọn ngôn ngữ ở mỗi trang, giúp thay đổi *req.query.lang*, từ đó thay đổi sang ngôn ngữ mới.
- Khi thay đổi ngôn ngữ đồng thời load lại trang, giúp thay đổi nội dung động đồng thời.

2. Ưu điểm và nhược điểm của phương án

a) Ưu điểm

- Dễ cài đặt, không dùng thư viện ngoài.
- Dễ tùy chỉnh, thay đổi mà không ảnh hưởng đến giao diện
- Thêm ngôn ngữ mới cần thêm json của ngôn ngữ đó và khai báo trong cài đặt *handlebars*.

b) Nhược điểm

- Khi file JSON quá lớn (~5,000 – 10,000 dòng), việc truy xuất có thể chậm và khó chỉnh sửa do có nhiều dòng cần xử lý.
- Có yêu cầu load lại trang, đôi lúc gây mất dữ liệu đang thao tác.

3. Các phương án dự phòng khi hệ thống lớn

c) Chia file json thành nhiều file nhỏ dựa theo views

- locales
 - o en
 - index.json
 - more.json
 - o vi
 - index.json
 - more.json

Ưu điểm:

- o Dễ quản lý từng file, nhanh chóng sửa ngôn ngữ theo views.
- o Tăng tốc độ xử lý: chỉ cần load file cần thiết theo từng trang.
- o Giảm rủi ro làm việc nhóm, tránh mất toàn bộ dữ liệu khi gộp.
- Nhược điểm:
 - o Phải đảm bảo đúng quy ước đặt tên
 - o Việc thêm ngôn ngữ mới chậm hơn một chút do phải chia thành nhiều file nhỏ.

d) Dùng thư viện hỗ trợ i18n chuyên sâu

- Một số thư viện phổ biến:
 - o I18n, i18next
 - o Node-polygot
 - o @ngx-translate (dành cho Angular)
- Ưu điểm (i18next):
 - o Lazy Load: Chỉ load phần dịch cần thiết tại thời điểm runtime.
 - o Namespaces: Tách dịch theo module hoặc màn hình, phù hợp chia nhỏ JSON.
 - o Fallback: Nếu thiếu khóa hoặc ngôn ngữ, sẽ fallback thông minh.
 - o Interpolation: Hỗ trợ truyền biến vào chuỗi dịch {{name}}.
 - o Plural: Hỗ trợ dịch số nhiều trong các ngôn ngữ khác.
 - o Async loading: Load file JSON từ API hoặc backend động.
- Nhược điểm:
 - o Yêu cầu cài đặt phức tạp.
 - o Không cần thiết, gây rắc rối khi dung lượng dịch vừa phải.

e) Kết nối với nền tảng quản lý dịch

- Một số nền tảng quản lý dịch:
 - o Locize

- Phrase
- POEditor
- **Ưu điểm:**
 - Cộng tác dễ dàng giữa dev, translator, tester.
 - Quản lý nhiều ngôn ngữ theo project.
 - Có thể kết nối CI/CD (deploy dịch song song với code).
- **Nhược điểm:**
 - Yêu cầu cấu hình phức tạp.
 - Một số nền tảng cần trả phí cho dự án lớn.
 - Cần cấu hình đồng bộ dữ liệu với repo hoặc server.
 - Chỉ phù hợp cho ứng dụng lớn đến rất lớn.

II. Phương án hỗ trợ đa ngôn ngữ cho các nội dung động

4. Mô tả phương án

- Tạo thêm bảng phụ (sub-table) cho bảng có trường con cần hỗ trợ ngôn ngữ. Tách trường con cần hỗ trợ ngôn ngữ vào bảng phụ:
 - Bảng chính:

module_id	module_code	credit	faculty_id	prerequisite_id	is_active
1					

- Bảng phụ:

module_id	language	name	description
1	en	-	-
1	vi	-	-

- Với các bảng không có trường thông tin không cần hỗ trợ ngôn ngữ, có thể thêm cột language vào bảng chính:

faculty_id	language	name
1	vi	CNTT
1	en	IT

- Với tên người và địa chỉ (ngôn ngữ mặc định Tiếng Việt), hệ thống chuyển thành chữ không dấu, bỏ các kí tự.

5. Các bảng và nội dung cần thay đổi

f) Bảng Module

- Bảng chính:

module_id	module_code	credits	faculty_id	prerequisite_id	is_active
1					

- Bảng phụ:

module_id	language	name	description
1	en	-	-
1	vi	-	-

g) Bảng Class

- Bảng chính:

class_id	module_id	academic_year	semester	instructor	max_students	classroom
1						

- Bảng phụ:

class_id	language	name	schedule
1	en		-
1	vi		-

h) Bảng Faculty – Khoa

faculty_id	language	name
------------	----------	------

i) Bảng Status – Trạng thái

status_id	language	name
-----------	----------	------

j) Thông tin sinh viên (student)

- Khi hiển thị thông tin, nếu ngôn ngữ không phải tiếng Việt, chuyển các thông tin:
 - o Tên
 - o Địa chỉ
 - o Quốc tịch

Thành từ không dấu, các ký tự Unicode thành ký tự ASCII tương ứng.

6. Ưu điểm và nhược điểm của phương án

k) Ưu điểm

- Giữ được bảng chính gọn nhẹ, chỉ chứa các dữ liệu cố định không thay đổi theo ngôn ngữ.
- Đơn giản, dễ hiểu: Mô hình dữ liệu rõ ràng, dễ thiết kế, dễ mở rộng khi thêm ngôn ngữ mới chỉ cần thêm bản ghi tương ứng.
- Tính linh hoạt cao, có thể áp dụng cho bất kỳ bảng nào chứa nội dung động cần đa ngôn ngữ
- Không phụ thuộc bên ngoài, không cần dùng thêm hệ thống dịch bên ngoài hoặc phức tạp.
- Dễ dàng đồng bộ hóa: Dữ liệu cho các ngôn ngữ khác nhau được lưu chung 1 bảng, dễ dàng so sánh, kiểm tra đồng bộ giữa các ngôn ngữ.
- Giảm thiểu dữ liệu trùng lặp, tiết kiệm dung lượng lưu trữ.

l) Nhược điểm

- Tăng kích thước dữ liệu: Bảng dữ liệu sẽ phình to lên theo số lượng ngôn ngữ, gây tốn bộ nhớ và chi phí truy vấn.
- Khó kiểm soát nhập liệu: Cần quy trình nhập liệu chặt chẽ để đảm bảo tất cả ngôn ngữ đều được nhập đủ và đồng bộ; nếu thiếu sẽ gây lỗi hoặc thiếu nội dung khi hiển thị.
- Phức tạp khi mở rộng: Nếu có nhiều bảng cần đa ngôn ngữ, phải sửa schema nhiều bảng hoặc áp dụng nhiều bảng song song cho các ngôn ngữ, gây phức tạp.
- Yêu cầu chỉnh sửa lớn trong mã nguồn: các đoạn định nghĩa mô hình, truy vấn dữ liệu.
- Tăng độ phức tạp trong truy vấn do phải join giữa bảng chính và bảng phụ
- Cần quy trình nhập liệu phức tạp hơn, đảm bảo đồng bộ dữ liệu đa ngôn ngữ trong bảng phụ.

7. Các phương án dự phòng khi hệ thống lớn

m) Sử dụng dịch vụ đa ngôn ngữ bên ngoài

- Sử dụng các API dịch thuật để hỗ trợ tự động dịch rồi trả cho người dùng.
- API của Google được tích hợp sẵn trên một số trình duyệt.
- **Ưu điểm:**
 - Tiết kiệm thời gian nhập liệu đa ngôn ngữ thủ công.
 - Dễ dàng mở rộng nhiều ngôn ngữ mới.
 - Tận dụng được công nghệ dịch tự động và chỉnh sửa thủ công.
 - Có thể dùng chung cho nội dung tĩnh.
- **Nhược điểm:**

- Chất lượng bản dịch tự động chưa chắc chính xác, có thể sai từ chuyên ngành; cần kiểm tra và xác thực.
- Có thể thay đổi nội dung bản dịch trong các lần dùng khác nhau; cần lưu lại một nội dung cố định.
- Phụ thuộc dịch vụ bên thứ ba, có thể có chi phí và rủi ro về bảo mật.
- Có thể tốn thời gian cho trình dịch.

III. Tài liệu tham khảo

- Google Translate Element:
https://www.w3schools.com/howto/howto_google_translate.asp