

# I. Coding Standards

## 1. Nguyên tắc chung.

Codebase tuân thủ các tiêu chuẩn được sử dụng phổ biến trong môi trường chuyên nghiệp để đảm bảo tính bảo trì, tính dễ đọc và sự nhất quán. Toàn bộ code phải tuân thủ các tiêu chuẩn này trước khi được hợp nhất vào nhánh chính.

## 2. Naming Conventions.

Về cách đặt tên variable và function.

- Sử dụng tên gọi có ý nghĩa, dễ hiểu và chỉ rõ mục đích.
- Đặt tên có format camelCase: `mailAddress`, `handleClick()`
- Đặt tên có format PascalCase cho class và constructor: `BaseIdentityDocument`, `Address`
- Đặt tên có format UPPER\_SNAKE\_CASE cho hằng số (constant) `VITE_API_BASE_URL`, `STATUS_TRANSITIONS`
- Về cách đặt tên file và thư mục:
- Sử dụng format camelCase: `multiSelect.tsx`, `usePrograms.ts`, `courseRoute.js`
  - Đối với server side, các file trong module quan trọng như controllers, routes, services sẽ được kèm hậu tố `<tên module>`. Ví dụ `class.service.js`, `program.controller.js`
  - Các file test sẽ chứa hậu tố `.test`, các file mock test sẽ chứa `.mock`. Chẳng hạn, `student.service.test.js`, `course.mock.js`

## 3. Code Structure and Formatting.

Indentation và Spacing.

- Sử dụng 2 dấu khoảng trắng cho indentation (không dùng tab).
- Luôn sử dụng dấu chấm phẩy ở cuối dòng code.

Comments.

- Có inline-comment cho những đoạn code phức tạp.

## 4. Error Handling.

- Luôn handle lỗi rõ ràng, không bao giờ bỏ qua.
- Luôn sử dụng khối lệnh try-catch trong các function, service quan trọng.
- Trả về thông báo lỗi có ý nghĩa với context.
- Nhất quán format lỗi.

## II. Cấu hình

DB\_HOST=< Postgres Database Connection >

DB\_USER=< Database User >

DB\_PASSWORD=< Database Password >

DB\_NAME=postgres

DB\_PORT=< Database Host >

PORT=< Server Port >

NODE\_ENV = <production | development >

## III. API Design

### 1. Nguyên tắc chung

- Thiết kế RESTful: API tuân thủ các tiêu chuẩn RESTful, sử dụng những HTTP method chuẩn (GET, POST, PUT, PATCH, DELETE).
- Resource-Oriented: API được thiết kế xoay quanh resource/đối tượng của hệ thống. /students, /classes, /programs, /faculties, /courses
- API endpoint luôn bắt đầu với tiền tố /api
- JSON format: Request và response thường áp dụng JSON format.

### 2. Base URL

Tất cả API endpoint thường bắt đầu từ cùng một đường dẫn: http://<host>:<port>/api.

Trong giai đoạn phát triển của team, đường dẫn được thống nhất là

http://localhost:3000/api

### 3. Request và Response Formats

- Content-Type:
  - Với những request có body (POST, PUT, PATCH), Content-Type header được chỉ định là application/json.
  - Với những request có đính kèm file, Content-Type header được chỉ định là multipart/form-data.
- JSON structure:
  - Response thông báo thành công:

```
{
```

```
"data": {  
  // Resource data or operation result  
}
```

- Response thông báo lỗi:

```
{  
  "statusCode": 400 // or 404, 500, etc.  
  "message": "ERROR MESSAGE IN ENGLISH",  
  "message_vi": "THÔNG BÁO LỖI BẰNG TIẾNG VIỆT"  
}
```

## 4. HTTP Status Codes

- 200 OK: Request được xử lý thành công.
- 201: Created: Một resource/object mới được tạo thành công.
- 400: Bad Request: Request chứa thông tin/ dữ liệu không hợp lệ.
- 409 Conflict: Request không thành công bởi vì dữ liệu hiện tại bị trùng lặp với trạng thái hiện tại của resource (duplicate).
- 404 Not Found: Resource được request không thể tìm thấy trong hệ thống.
- 500 Internal Server Error: Lỗi chung xảy ra ở server.

## 5. Error Handling

Response chứa lỗi cần chứa đầy đủ 3 thông tin: error code, message (mô tả lỗi được viết bằng Tiếng Anh), message\_vi (mô tả lỗi được viết bằng tiếng Việt). Lỗi runtime được xảy ra phải được xử lý qua Error Handler được define sẵn.

## 6. Pagination, Filtering, Sorting

- Pagination:
  - Sử dụng query parameters, chẳng hạn `?page=1&limit=10`.
  - Responses chứa pagination metadata: total (tổng số lượng resource có trong hệ thống).
- Filtering:
  - Sử dụng query parameter có tên là các field trùng với field được define trong model? Ví dụ: fullName, programCode
- Sorting:
  - Sử dụng query parameter: `?sortBy=studentCode&sortOrder=desc`.

## IV. Web API Documentation

[Upcoming]