

TRƯỜNG CAO ĐẲNG FPT POLYTECHNIC CƠ SỞ CẦN THƠ

BỘ MÔN CÔNG NGHỆ THÔNG TIN

ĐỀ THI PHP 2 – LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Thông tin chung

- Thời gian làm bài: 60 phút
- Quy định:
 - Sinh viên **không được sử dụng internet** trong quá trình làm bài.
 - Sinh viên sử dụng lại các file code mẫu để hoàn thiện bài thi.
 - Đề thi gồm **1 trang** với 8 câu hỏi, tổng điểm 10.
 - Sinh viên cần viết code rõ ràng, đúng cú pháp PHP, và thêm chú thích (comment) để giải thích logic.
 - Các file liên quan: Validatable.php, Model.php, Category.php, Viewer.php, HomeController.php.
 - Sử dụng **PDO** để thao tác với cơ sở dữ liệu.
 - Giả sử có sẵn hàm redirect(\$url) để chuyển hướng trang.

Cơ sở dữ liệu

Bảng categories trong cơ sở dữ liệu có cấu trúc như sau:

- id: int, khóa chính, tự động tăng.
- name: varchar(255), tên danh mục, **bắt buộc** không được rỗng.
- description: text, mô tả danh mục, có thể để trống.
- status: varchar(50), trạng thái danh mục (ví dụ: "active", "inactive").
- priority: int, mức độ ưu tiên, **phải lớn hơn hoặc bằng 0**, mặc định là 0.

Câu hỏi

Câu 1 (1 điểm)

Viết một **interface** có tên Validatable với một phương thức trừu tượng validate(array \$data): bool. Phương thức này dùng để kiểm tra tính hợp lệ của dữ liệu đầu vào.

Yêu cầu:

- Interface được khai báo trong file Validatable.php.
- Phương thức validate nhận một mảng \$data và trả về true nếu dữ liệu hợp lệ, false nếu không.

Câu 2 (1 điểm)

Trong file Model.php, giải thích vai trò của thuộc tính protected \$table.

Yêu cầu:

- Viết đoạn code khai báo thuộc tính \$table.
- Thêm chú thích (comment) giải thích vai trò của thuộc tính này.

Câu 3 (1 điểm)

Trong file Model.php, giải thích vai trò của phương thức __construct().

Yêu cầu:

- Viết đoạn code khai báo phương thức __construct() để khởi tạo kết nối cơ sở dữ liệu sử dụng PDO.
- Thêm chú thích (comment) giải thích vai trò của phương thức này.

Câu 4 (1 điểm)

Trong class Category (file Category.php), viết phương thức getTableName(): string để trả về giá trị của thuộc tính \$table (giá trị là "categories").

Yêu cầu:

- Phương thức trả về kiểu string.
- Thêm chú thích (comment) giải thích chức năng của phương thức.

Câu 5 (1.5 điểm)

Trong file Viewer.php, giải thích chức năng của phương thức render(string \$template, array \$data = []).

Yêu cầu:

- Viết lại phương thức này với chú thích (comment) rõ ràng.
- Phương thức nhận tham số \$template (đường dẫn tới file giao diện) và \$data (mảng dữ liệu để truyền vào giao diện).
- Sử dụng hàm extract() để trích xuất dữ liệu từ mảng \$data và hàm require để nạp file giao diện.

Câu 6 (1.5 điểm)

Trong class Category (file Category.php), viết phương thức validate(array \$data): bool để kiểm tra tính hợp lệ của dữ liệu đầu vào.

Yêu cầu:

- Kiểm tra:
 - Trường name **không được rỗng**.
 - Trường priority **phải là số nguyên** và **lớn hơn hoặc bằng 0**.
- Trả về true nếu dữ liệu hợp lệ, false nếu không.
- Thêm chú thích (comment) giải thích logic kiểm tra.

Câu 7 (2 điểm)

Trong class Category (file Category.php), sửa phương thức store(\$data) để kiểm tra dữ liệu bằng phương thức validate() trước khi lưu vào cơ sở dữ liệu.

Yêu cầu:

- Nếu dữ liệu không hợp lệ, ném ra ngoại lệ Exception với thông báo **"Invalid data"**.
- Nếu hợp lệ, lưu dữ liệu vào bảng categories sử dụng PDO.
- Phương thức nhận tham số \$data là một mảng chứa các trường name, description, status, priority.
- Thêm chú thích (comment) giải thích logic.

Câu 8 (2 điểm)

Trong file HomeController.php, sửa phương thức xuLyCategoryEdit() để xử lý việc chỉnh sửa danh mục.

Yêu cầu:

- Kiểm tra dữ liệu đầu vào bằng phương thức validate() của class Category.
- Nếu dữ liệu hợp lệ, gọi phương thức update() của class Category để cập nhật danh mục trong cơ sở dữ liệu.
- Nếu dữ liệu không hợp lệ, trả về thông báo lỗi **"Invalid data"** và render lại form chỉnh sửa (sử dụng Viewer::render()) với dữ liệu cũ.
- Nếu cập nhật thành công, sử dụng hàm redirect(\$url) để chuyển hướng về trang danh sách danh mục (/).
- Thêm chú thích (comment) giải thích logic.