**KHOA CNTT – TRƯỜNG ĐẠI HỌC THỦY LỢI**

Học phần : **Công nghệ Web**

Khóa : **K63**

**BÀI TẬP THỰC HÀNH**

**Buổi 01**

**Mục tiêu:**

* Làm quen với việc sử dụng **git/github** (gitlab/bitbucket) trong việc Quản lý phiên bản & hợp tác Lập trình nhóm.
* Ôn tập các câu lệnh SQL
* Thực hành PHP – MySQL với phương pháp PDO

1. **Sinh viên thành lập nhóm từ 3 – 4 người và thực hiện các yêu cầu:**

* Cài đặt **git** trên máy (Tất cả các thành viên)
* Đăng kí tài khoản **github** (nếu sử dụng github và các thành viên trong nhóm đều chưa có tài khoản). Sử dụng tài khoản có sẵn (nếu một trong các thành viên đã có)
* Phân công trưởng nhóm và tạo dự án đặt tên **CSE485\_2023**; thực hiện invite các thành viên trong nhóm vào Dự án.

**PHẦN HỢP TÁC (THỰC HIỆN CHUNG**)

1. Tạo một Cơ sở dữ liệu có tên **BTTH01\_CSE485**
2. Sử dụng **phpMyAdmin** hoặc **MySQL workbench** để làm việc với CSDL trên và tạo một tệp tin SQL theo các bước sau:

*(Lưu ý: Bài tập phải được lưu với tên “BTTH01\_CSE485\_ex.sql”, nếu không sẽ không được chấm)*

1. Tạo các Bảng trong CSDL nói trên (*Có dữ liệu mẫu cho kèm theo để minh họa*) (**6 đ**)

Tác giả (**tacgia**)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| ma\_tgia | int unsigned | no | primary key |  |  |
| ten\_tgia | varchar(100) | no |  |  |  |
| hinh\_tgia | varchar(100) | yes |  |  |  |

Bài viết (**baiviet**)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| ma\_bviet | int unsigned | no | primary key |  |  |
| tieude | varchar(200) | no |  |  |  |
| ten\_bhat | varchar(100) | no |  |  |  |
| ma\_tloai | int unsigned | no | foreign key |  |  |
| tomtat | text | no |  |  |  |
| noidung | text | yes |  |  |  |
| ma\_tgia | int unsigned | no | foreign key |  |  |
| ngayviet | datetime | no |  | Ngày hiện tại từ HT |  |
| hinhanh | varchar(200) | yes |  |  |  |

Thể loại (**theloai**)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| ma\_tloai | int unsigned | no | primary key |  |  |
| ten\_tloai | varchar(50) | no |  |  |  |

***Lưu ý***: Sinh viên sử dụng dữ liệu trong các tệp *theloai-data.sql, tacgia-data.sql, baiviet-data.sql*

1. Sinh viên tự xác định và thiết lập các loại ràng buộc cho phù hợp với yêu cầu (**2 đ**)
2. Tìm và sửa lỗi sai về thiết kế (nếu có, **5 đ)**
3. ***Thực hiện các yêu cầu truy vấn sau:***
   1. Liệt kê các bài viết về các bài hát thuộc thể loại Nhạc trữ tình (**2 đ**)

SELECT tieude FROM baiviet INNER JOIN theloai ON baiviet.ma\_tloai = theloai.ma\_tloai WHERE theloai.ten\_tloai = N'Nhạc trữ tình';

* 1. Liệt kê các bài viết của tác giả “Nhacvietplus” (**2 đ**)

SELECT tieude FROM baiviet INNER JOIN tacgia ON baiviet.ma\_tgia = tacgia.ma\_tgia WHERE tacgia.ten\_tgia = N'Nhacvietplus';

* 1. Liệt kê các thể loại nhạc chưa có bài viết cảm nhận nào. (**2 đ**)

SELECT ten\_tloai FROM theloai LEFT JOIN baiviet ON theloai.ma\_tloai = baiviet.ma\_tloai WHERE baiviet.ma\_bviet IS null;

* 1. Liệt kê các bài viết với các thông tin sau: mã bài viết, tên bài viết, tên bài hát, tên tác giả, tên thể loại, ngày viết. (**2 đ**)

SELECT

baiviet.ma\_bviet AS 'Mã bài viết',

baiviet.tieude AS 'Tên bài viết',

baiviet.ten\_bhat AS 'Tên bài hát',

tacgia.ten\_tgia AS 'Tên tác giả',

theloai.ten\_tloai AS 'Tên thể loại',

baiviet.ngayviet AS 'Ngày viết'

FROM baiviet, theloai, tacgia where baiviet.ma\_tgia = tacgia.ma\_tacgia and baiviet.ma\_tloai = theloai.ma\_tloai

* 1. Tìm thể loại có số bài viết nhiều nhất (**2 đ**)

SELECT

theloai.ten\_tloai AS 'Thể loại',

COUNT(baiviet.ma\_bviet) AS 'Số bài viết'

FROM

theloai

LEFT JOIN baiviet ON theloai.ma\_tloai = baiviet.ma\_tloai

GROUP BY

theloai.ten\_tloai

ORDER BY

COUNT(baiviet.ma\_bviet) DESC

LIMIT 1;

* 1. Liệt kê 2 tác giả có số bài viết nhiều nhất (**2 đ**)

SELECT

tacgia.ten\_tgia AS 'Tác giả',

COUNT(baiviet.ma\_bviet) AS 'Số bài viết'

FROM

tacgia

LEFT JOIN baiviet ON tacgia.ma\_tgia = baiviet.ma\_tgia

GROUP BY

tacgia.ten\_tgia

ORDER BY

COUNT(baiviet.ma\_bviet) DESC

LIMIT 2;

* 1. Liệt kê các bài viết về các bài hát có tựa bài hát chứa 1 trong các từ “yêu”, “thương”, “anh”, “em” (**2 đ**)

SELECT

ma\_bviet AS 'Mã bài viết',

tieude AS 'Tên bài viết',

ten\_bhat AS 'Tên bài hát'

FROM

baiviet

WHERE

ten\_bhat LIKE '%yêu%' OR

ten\_bhat LIKE '%thương%' OR

ten\_bhat LIKE '%anh%' OR

ten\_bhat LIKE '%em%';

* 1. Liệt kê các bài viết về các bài hát có tiêu đề bài viết hoặc tựa bài hát chứa 1 trong các từ “yêu”, “thương”, “anh”, “em” (**2 đ**)

SELECT

ma\_bviet AS 'Mã bài viết',

tieude AS 'Tên bài viết',

ten\_bhat AS 'Tên bài hát'

FROM

baiviet

WHERE

ten\_bhat LIKE '%yêu%' OR

ten\_bhat LIKE '%thương%' OR

ten\_bhat LIKE '%anh%' OR

ten\_bhat LIKE '%em%';

* 1. Tạo 1 view có tên vw\_Music để hiển thị thông tin về Danh sách các bài viết kèm theo Tên thể loại và tên tác giả (**2 đ**)

CREATE VIEW vw\_Music AS SELECT bv.ma\_bviet, bv.tieude, bv.ten\_bhat, bv.tomtat, bv.noidung, bv.ngayviet, tl.ten\_tloai, tg.ten\_tgia FROM baiviet bv INNER JOIN theloai tl ON bv.ma\_tloai = tl.ma\_tloai INNER JOIN tacgia tg ON bv.ma\_tgia = tg.ma\_tgia;

* 1. Tạo 1 thủ tục có tên sp\_DSBaiViet với tham số truyền vào là Tên thể loại và trả về danh sách Bài viết của thể loại đó. Nếu thể loại không tồn tại thì hiển thị thông báo lỗi. (**2 đ**)

CREATE PROCEDURE sp\_DSBaiViet @TenTheLoai NVARCHAR(255) AS BEGIN IF EXISTS (SELECT 1 FROM theloai WHERE ten\_tloai = @TenTheLoai) BEGIN SELECT bv.ma\_bviet, bv.tieude, bv.ten\_bhat, bv.tomtat, bv.noidung, bv.ngayviet, tl.ten\_tloai, tg.ten\_tgia FROM baiviet bv INNER JOIN theloai tl ON bv.ma\_tloai = tl.ma\_tloai INNER JOIN tacgia tg ON bv.ma\_tgia = tg.ma\_tgia WHERE tl.ten\_tloai = @TenTheLoai; END ELSE BEGIN RAISEERROR('Thể loại không tồn tại', 16, 1); END END;

* 1. Thêm mới cột SLBaiViet vào trong bảng theloai. Tạo 1 trigger có tên tg\_CapNhatTheLoai để khi thêm/sửa/xóa bài viết thì số lượng bài viết trong bảng theloai được cập nhật theo. (**2 đ**)

CREATE TRIGGER tg\_CapNhatTheLoai

ON baiviet

AFTER INSERT, UPDATE, DELETE

AS

BEGIN

IF OBJECT\_ID('tempdb..#TempTable') IS NOT NULL

DROP TABLE #TempTable;

CREATE TABLE #TempTable

(

MaTheLoai INT,

SoLuong INT

);

-- Cập nhật số lượng bài viết cho từng thể loại

INSERT INTO #TempTable (MaTheLoai, SoLuong)

SELECT

ma\_tloai,

COUNT(\*) AS SoLuong

FROM

baiviet

GROUP BY

ma\_tloai;

-- Update SLBaiViet trong bảng theloai

UPDATE tl

SET SLBaiViet = t.SoLuong

FROM theloai tl

INNER JOIN #TempTable t ON tl.ma\_tloai = t.MaTheLoai;

DROP TABLE #TempTable;

END;

* 1. Bổ sung thêm bảng **Users** để lưu thông tin Tài khoản đăng nhập và sử dụng cho chức năng Đăng nhập/Quản trị trang web. (**5 đ**)

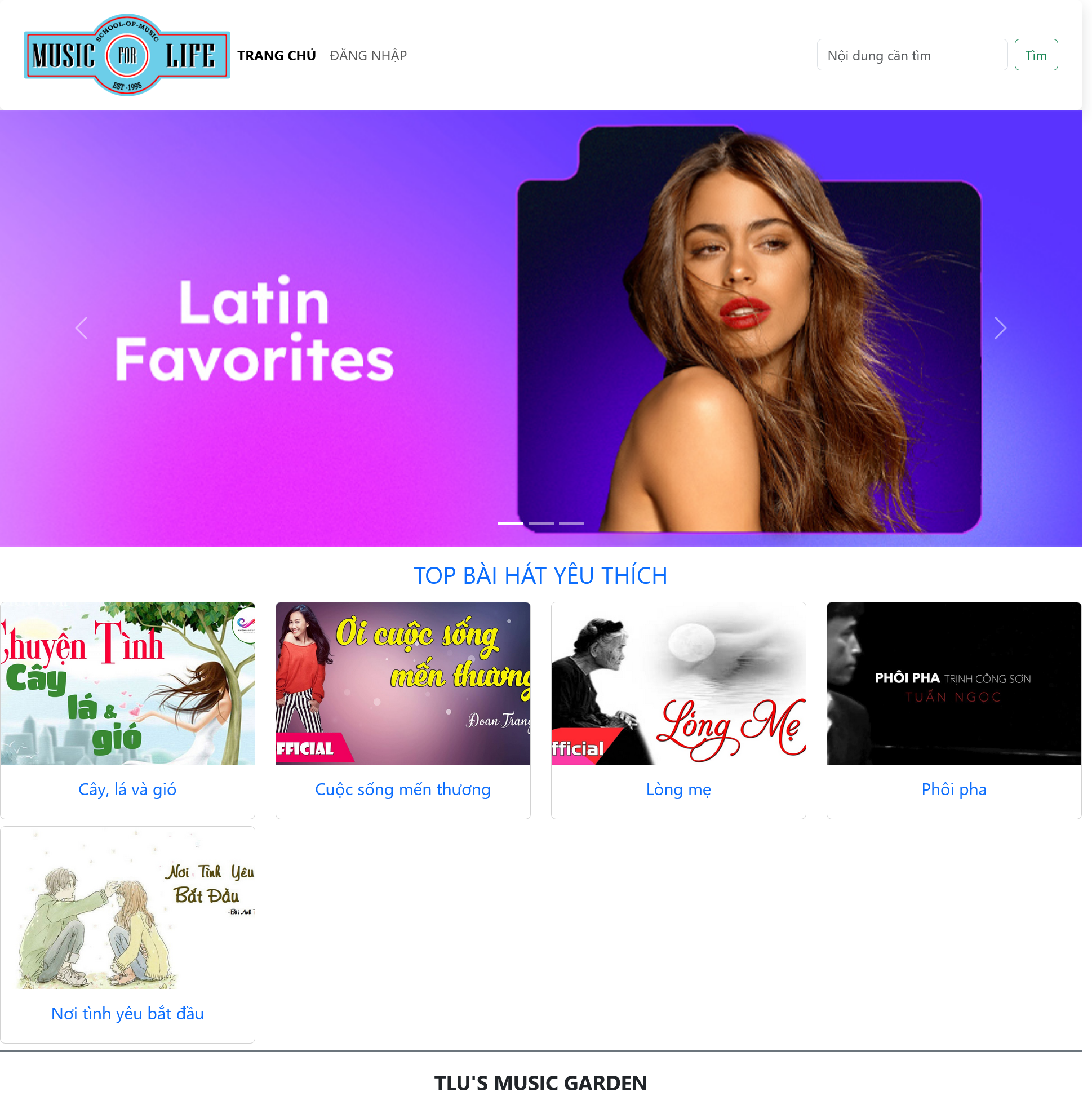
***Yêu cầu:***

* Sinh viên thực hiện tạo thư mục **database** trong dự án, xuất dữ liệu với tên mặc định, sao chép 2 tệp tin (dữ liệu *BTTH01\_CSE485.sql* + bài tập *BTTH01\_CSE485\_ex.sql*) vào thư mục **database** và đồng bộ dự án lên github. Sinh viên cần thực hiện chính xác yêu cầu, nếu không sẽ không được chấm điểm.
* Dữ liệu được gửi và sử dụng cho tất cả các thành viên trong nhóm.
* Các thành viên sử dụng MẪU dự án Web cho sẵn (**BTTH1\_template.zip**) và phân chia công việc để thực hiện NHIỆM VỤ RIÊNG bên dưới đây:

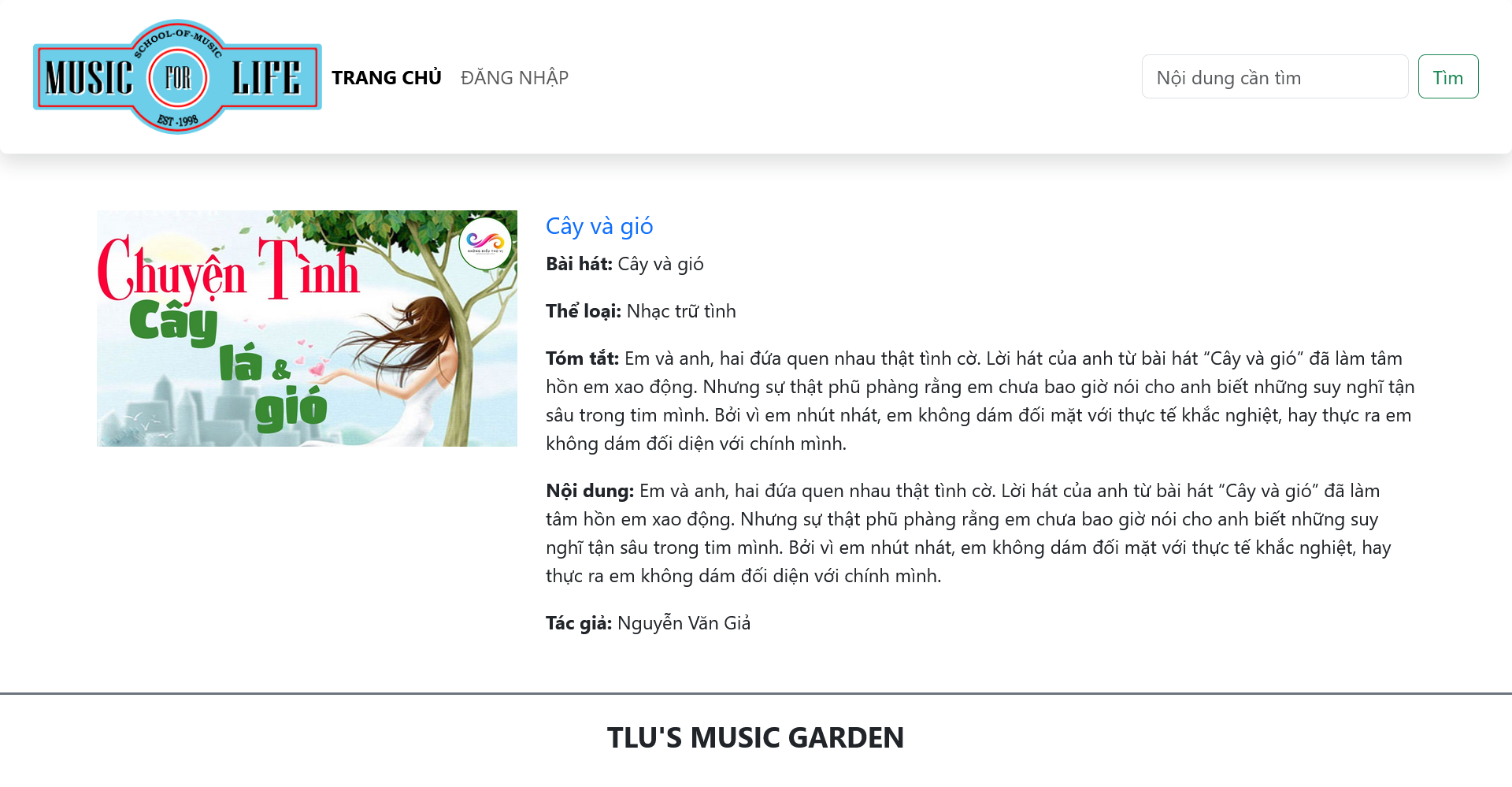
**PHẦN NHIỆM VỤ RIÊNG (THỰC HIỆN THEO PHÂN CÔNG CỦA TRƯỞNG NHÓM**)

1. Sử dụng kỹ thuật lập trình với PDO hoàn thiện theo các yêu cầu chi tiết phía dưới **(60 đ)**

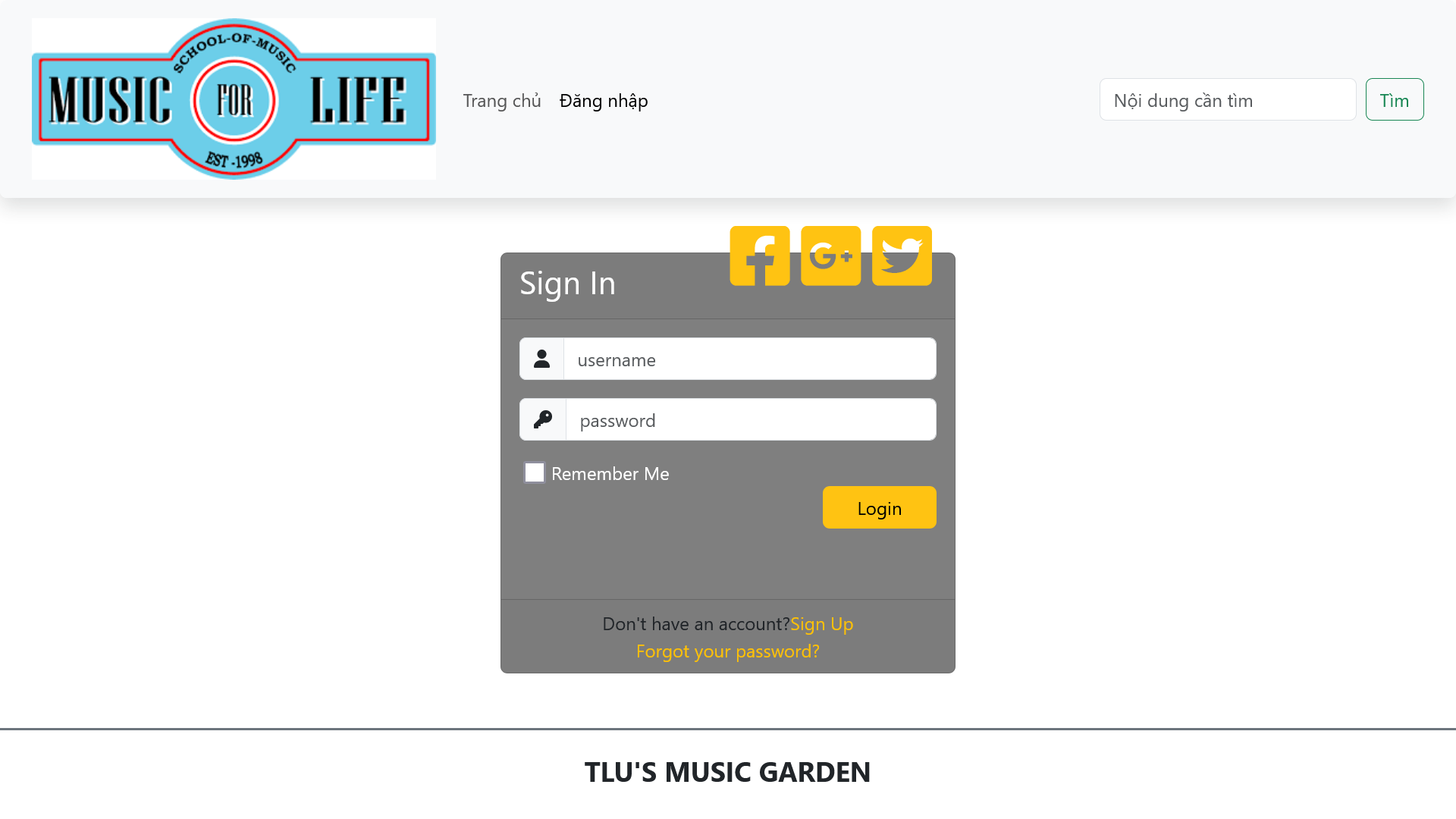
*index.php (trang chủ)*



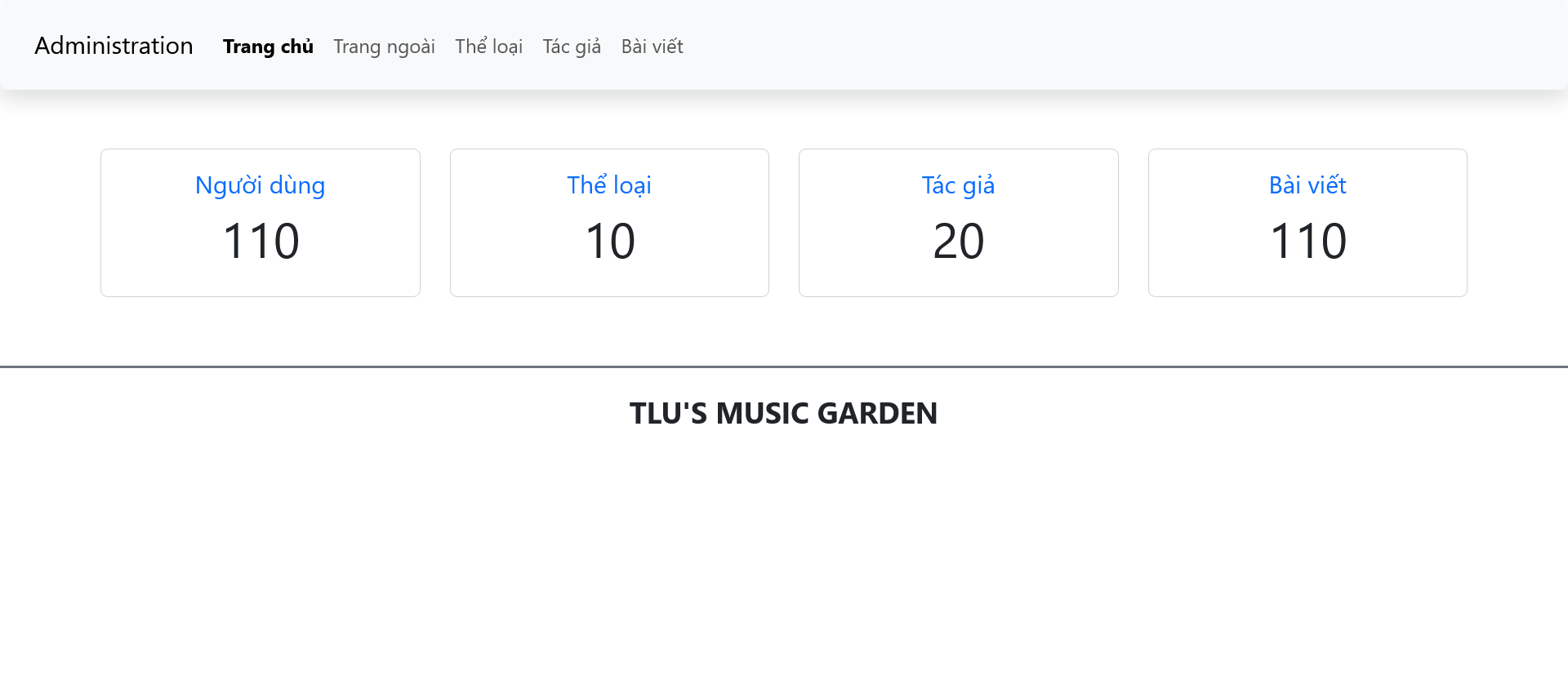
*detail.php (Chi tiết bài viết cảm nhận về bài hát)*



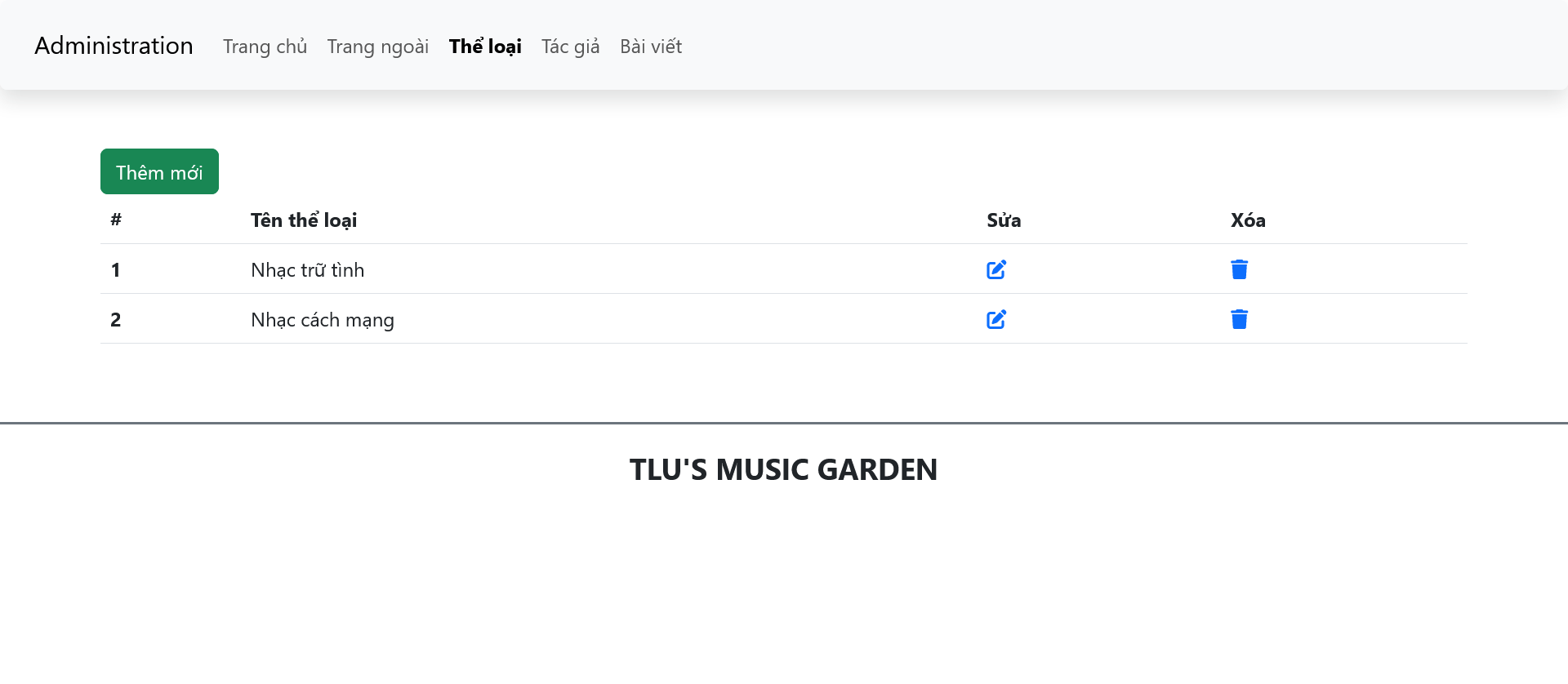
*login.php (trang đăng nhập)*



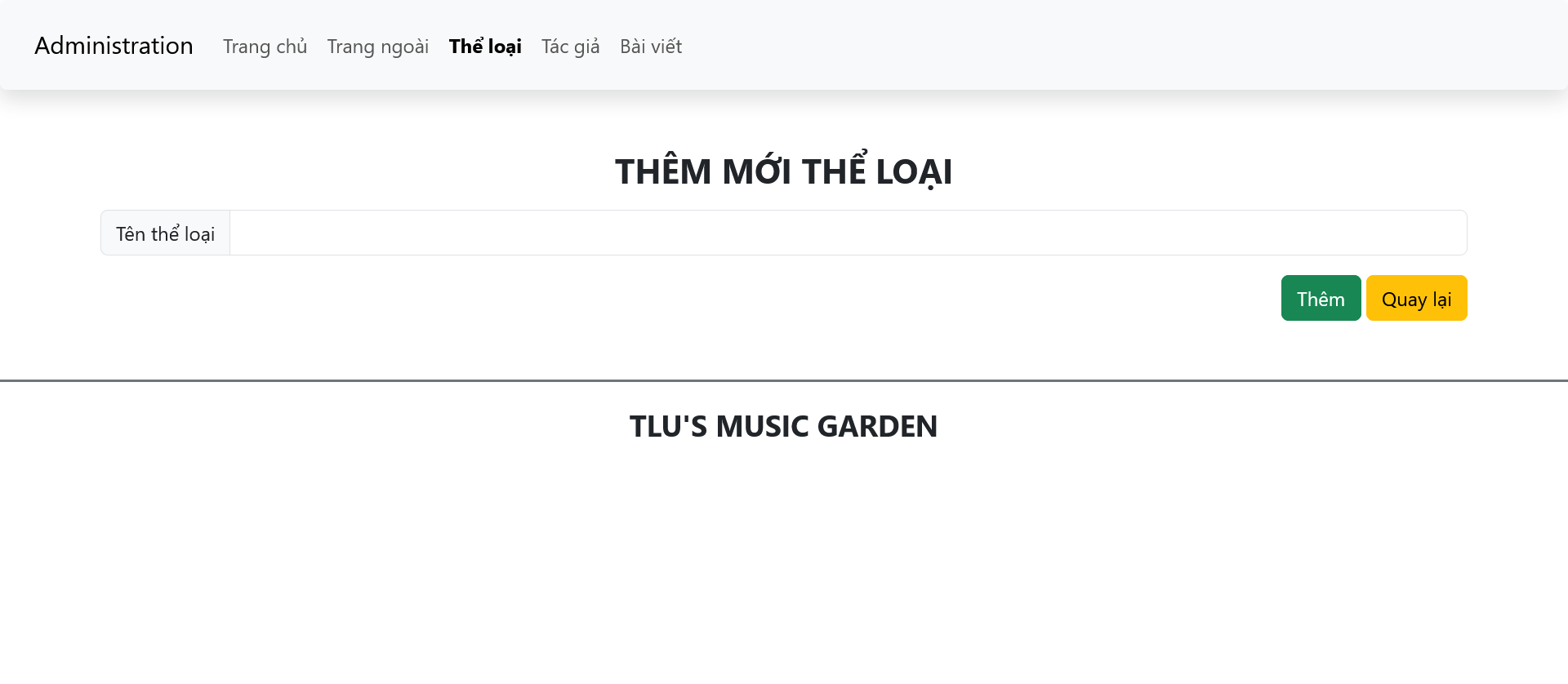
*admin/index.php (Trang chủ - vùng quản trị)*



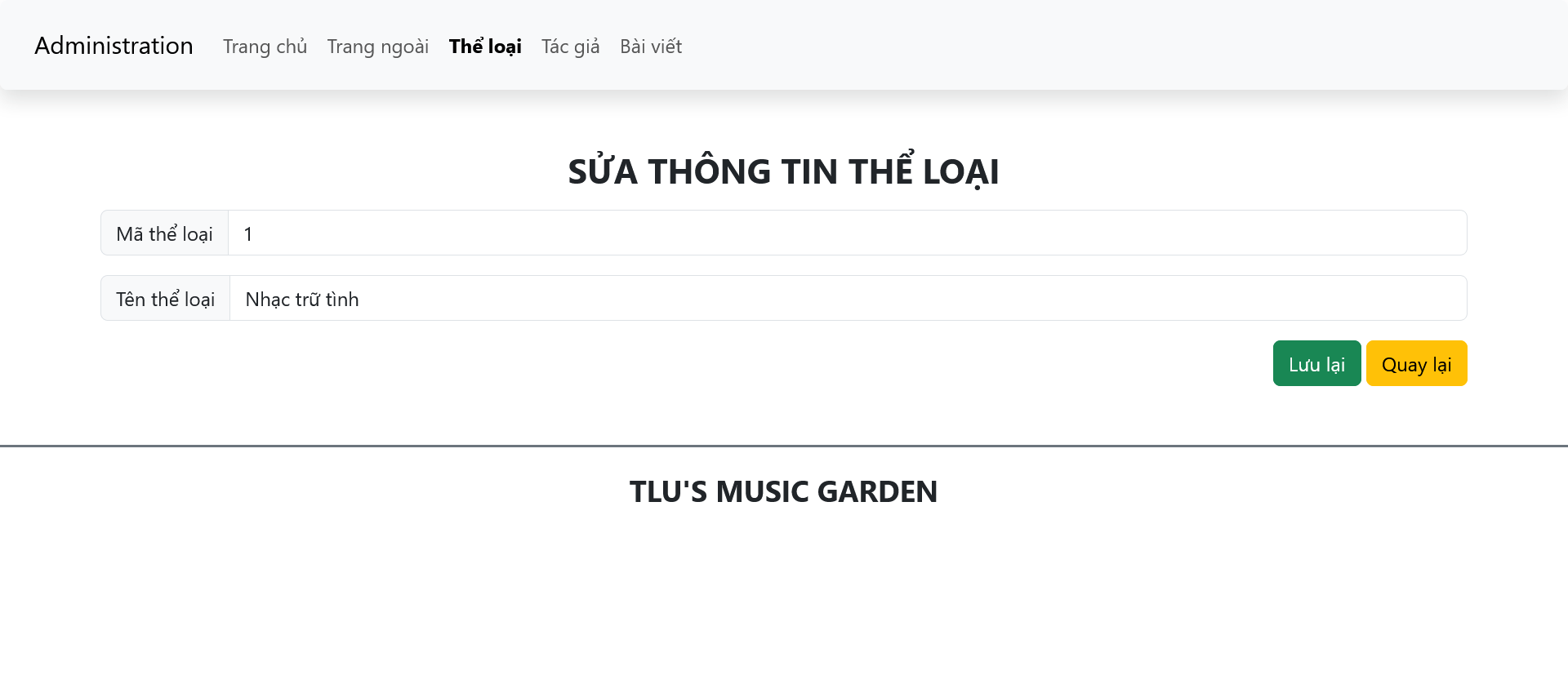
*admin/category.php (trang quản lý Thể loại – vùng Quản trị)*



*admin/add\_category.php (Trang thêm mới thể loại – vùng quản trị)*



*admin/edit\_category.php (trang sửa thông tin thể loại – vùng quản trị)*



***Lưu ý***: Với cách làm tương tự, sinh viên tự thiết kế cho phù hợp với Quản lý Tác giả, Bài viết

1. **Yêu cầu 1**: Sinh viên tùy ý chỉnh sửa lại giao diện tổng thể sao cho hợp lý hơn về phối màu và có trải nghiệm người dùng tốt hơn. Sinh viên tự xác định và chia tách/tái sử dụng code một cách hợp lý. **(+ 5 – 10 đ)**
2. **Yêu cầu 2:** Các thành viên trong nhóm Rẽ nhánh trên Github để thực hiện phần việc được phân công. Trưởng nhóm chịu trách nhiệm quản lý và merge code vào nhánh chính.
3. **Yêu cầu 3:** Sinh viên thêm giao diện tương tự còn thiếu cho các Mục còn lại. **(10 đ)**
4. **Yêu cầu 4**: Sinh viên hoàn thiện các chức năng (Đăng nhập; Thêm/Sửa/Xóa Thể loại; Thêm/Sửa/Xóa Tác giả; Thêm/Sửa/Xóa Bài viết) **(Mỗi chức năng 5 đ)**

**HẾT**