

TRƯỜNG ĐẠI HỌC HẠ LONG  
KHOA CÔNG NGHỆ THÔNG TIN



**BÁO CÁO KẾT THÚC HỌC PHẦN**  
**MÔN: PHÁT TRIỂN PHẦN MỀM MÃ NGUỒN MỞ**  
**Đề tài: Xây dựng hệ thống phần mềm mã nguồn mở quản lý**  
**Tour du lịch tại một công ty du lịch**

**Giảng viên hướng dẫn:** TS. Nguyễn Văn Thắm  
**Lớp:** KHMTK6B

Sinh viên thực hiện	Điểm
1. Phạm Minh Thiên	
2. Trần Xuân Thành	
3. Đỗ Văn Thành	

**Giảng viên chấm 1**

**Giảng viên chấm 2**

Quảng Ninh, tháng 06 năm 2024

## **LỜI CẢM ƠN**

Chúng em muốn bày tỏ lòng biết ơn sâu sắc đối với sự hướng dẫn và sự tận tâm mà Thầy Nguyễn Văn Thắm đã mang đến trong quá trình giảng dạy môn Phát Triển Phần Mềm Mã Nguồn Mở. Và chúng em cũng vô cùng cảm ơn Thầy đã nhiệt tình hướng dẫn chúng em hoàn thành tốt bài báo cáo này.

Chúng em tin rằng, những kiến thức và kỹ năng của môn học Phát Triển Phần Mềm Mã Nguồn Mở mà chúng em đã học được dưới sự hướng dẫn của Thầy Nguyễn Văn Thắm không chỉ là nền tảng vững chắc cho sự phát triển nghề nghiệp mà còn là công cụ quan trọng giúp chúng em vượt qua những thách thức.

Em cũng xin cảm ơn các bạn trong lớp KHMT K6B đã luôn sẵn lòng hỗ trợ, chia sẻ kinh nghiệm và cùng nhau vượt qua những khó khăn trong quá trình thực hiện đề tài.

Một lần nữa, chúng em chân thành cảm ơn và kính chúc thầy luôn mạnh khỏe, hạnh phúc và thành công trong công việc giảng dạy cũng như trong cuộc sống cá nhân của mình.

## MỤC LỤC

LỜI CẢM ƠN .....	1
CHƯƠNG 1.....	4
TỔNG QUAN VỀ BÀI TOÁN XÂY DỰNG HỆ THỐNG PHẦN MỀM MÃ NGUỒN MỞ QUẢN LÝ TOUR DU LỊCH TẠI CÔNG TY DU LỊCH .....	4
1.1. Trình bày tổng quan bài toán kinh doanh điện thoại .....	4
1.1.1. Giới thiệu.....	4
1.1.2. Mục tiêu.....	4
1.1.3. Chức năng của hệ thống.....	5
1.2. Trình bày các giải pháp thực hiện .....	5
CHƯƠNG 2.....	6
MÔ HÌNH CHỨC NĂNG VÀ MÔ HÌNH DỮ LIỆU BÀI TOÁN QUẢN LÝ TOUR DU LỊCH.....	6
2.1. Mô hình chức năng.....	6
2.2. Mô hình cơ sở dữ liệu của bài quản lý tour du lịch .....	6
2.2.1. Mô hình thực thể liên hệ .....	6
2.2.2. Các bảng dữ liệu.....	7
2.2.3. Mô hình quan hệ.....	9
2.3. Cấu trúc mô hình MVC trên mã nguồn mở Laravel .....	9
2.3.1. Cấu trúc chung .....	9
2.3.2. Cấu hình các route.....	10
2.2.3. Cấu hình cơ sở dữ liệu .....	16
2.2.4. Tầng Controller .....	19
2.2.5. Tầng Model .....	35
2.2.6. Tầng View .....	39
CHƯƠNG 3.....	49
CÀI ĐẶT VÀ CÁC GIAO DIỆN WEBSITE.....	49

3.1. Giao diện trang chủ .....	49
3.2. Giao diện các trang cho nhà quản trị.....	51
3.3. Giao diện các trang cho người dùng .....	57
KẾT LUẬN .....	60
TÀI LIỆU THAM KHẢO.....	61

## **CHƯƠNG 1**

### **TỔNG QUAN VỀ BÀI TOÁN XÂY DỰNG HỆ THỐNG PHẦN MỀM MÃ NGUỒN MỞ QUẢN LÝ TOUR DU LỊCH TẠI CÔNG TY DU LỊCH**

#### **1.1. Trình bày tổng quan bài toán kinh doanh điện thoại**

##### **1.1.1. Giới thiệu**

Ngày nay, do nhu cầu du lịch của người dân phát triển, nhiều công ty du lịch được thành lập để đáp ứng yêu cầu du lịch rất đa dạng của người dân. Một trong nhiệm vụ của hoạt động du lịch là tạo ra các tour du lịch theo hành trình khác nhau, có giá khác nhau, có tiện ích và sự lôi cuốn khác nhau và quảng bá nó để thu hút mọi người. Khi có người đăng ký thì nhanh chóng làm thủ tục và tiến hành triển khai thực hiện nó một cách hiệu quả. Do sự cạnh tranh rất lớn, ngoài việc cung cấp những tour hấp dẫn, có các dịch vụ tốt, thì giảm chi phí là một nhu cầu rất lớn để cạnh tranh về giá thành.

Do sự phát triển của nhiều cơ sở dịch vụ khác nhau như nhà hàng, khách sạn, các công ty vận tải cung cấp phương tiện, các đơn vị cung cấp hướng dẫn viên,...nên nhiệm vụ chủ yếu của các ty du lịch là nắm vững được các thông tin về dịch vụ và danh lam thắng cảnh ở các địa phương, từ đó tổ chức tốt các tour, sử dụng tối đa các dịch vụ có được, không cần phải có một tổ chức đông người, công kênh cũng có thể kinh doanh thành công.

Vì thế, các công ty phải ứng dụng công nghệ thông tin để hoạt động hiệu quả. Hệ thống phần mềm quản lý tour du lịch đóng vai trò quan trọng trong việc giúp các công ty lý hành tối ưu hóa quy trình kinh doanh, cải thiện dịch vụ khách hàng và nâng cao hiệu suất làm việc.

Việc phát triển hệ thống này dưới dạng mã nguồn mở không chỉ mang lại tính linh hoạt và khả năng tùy chỉnh cao, mà còn tận dụng được sức mạnh của cộng đồng phát triển để liên tục cải tiến và mở rộng tính năng.

##### **1.1.2. Mục tiêu**

Hệ thống phần mềm mã nguồn mở quản lý tour du lịch được đề xuất nhằm mục tiêu:

Tự động hóa quy trình quản lý: Giảm thiểu sự can thiệp của con người trong các quy trình như đặt tour, thanh toán và cập nhật thông tin.

Tăng cường độ chính xác và bảo mật: Đảm bảo thông tin được lưu trữ và xử lý một cách an toàn, giảm thiểu rủi ro mất mát hoặc sai lệch dữ liệu.

Cải thiện trải nghiệm khách hàng: Cung cấp các dịch vụ nhanh chóng, chính xác và tiện lợi cho khách hàng.

### **1.1.3. Chức năng của hệ thống**

Quản lý tour du lịch: Theo dõi chi tiết thông tin về các tour du lịch bao gồm lịch trình, giá cả, địa điểm và số lượng chỗ trống.

Quản lý khách hàng: Lưu trữ và quản lý thông tin, tài khoản khách hàng.

Quản lý Booking: Theo dõi đặt chỗ từ khách hàng và quản lý thông tin chi tiết về các booking.

Quản lý Menu: Quản lý menu của trang website

### **1.2. Trình bày các giải pháp thực hiện**

- **Quản lý tour du lịch:**

Xây dựng một giao diện quản lý cho phép người quản trị thêm mới, chỉnh sửa và xóa các lịch trình tour du lịch.

Cho phép người quản trị quản lý giá cả của các tour

Quản lý thông tin về các điểm đến trong tour.

- **Quản lý Khách hàng:**

Lưu trữ thông tin khách hàng như tên, địa chỉ, số điện thoại và email để quản lý quan hệ khách hàng.

Lưu thông tin tài khoản của khách hàng.

- **Quản lý Booking:**

Thiết kế giao diện đặt tour trực tuyến cho phép khách hàng chọn tour, nhập thông tin cá nhân và đặt chỗ.

Lưu trữ thông tin chi tiết về mỗi booking bao gồm thông tin cá nhân của khách hàng, tour đặt chỗ, ngày khởi hành, số lượng người tham gia và các yêu cầu đặc biệt.

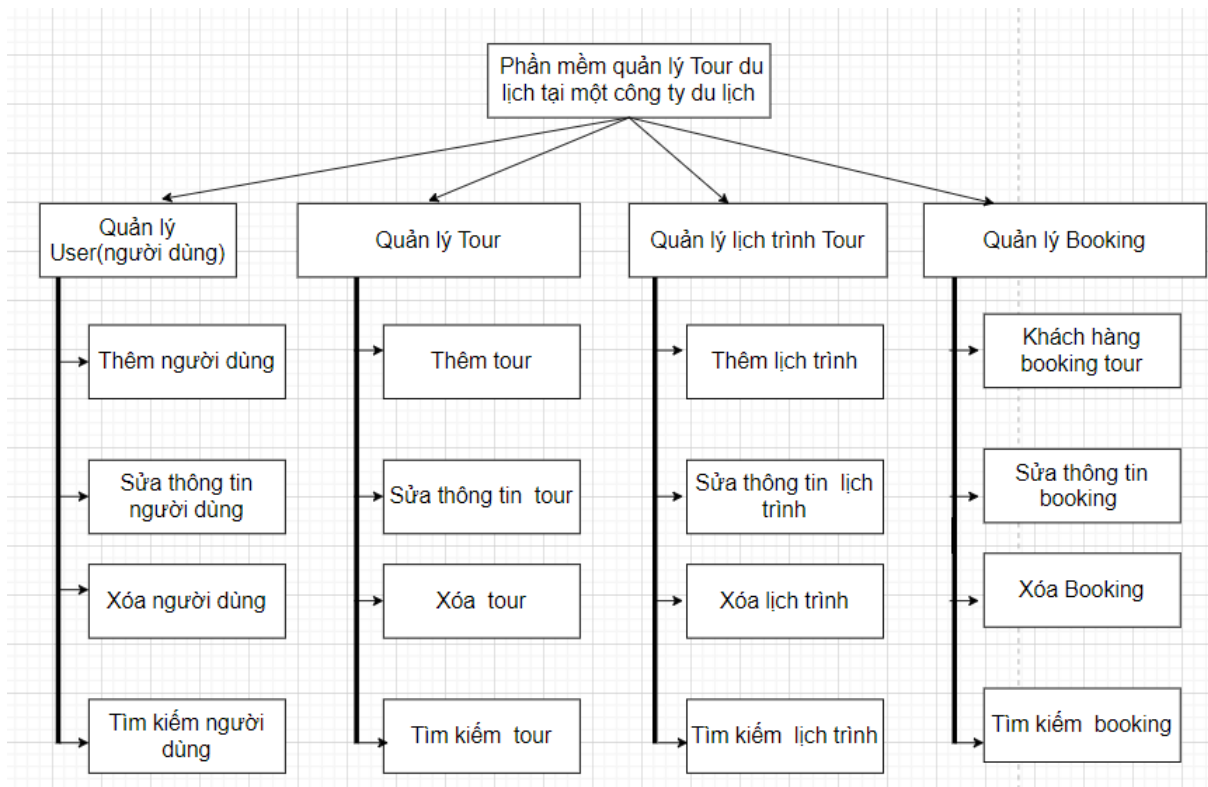
- **Quản lý Menu:**

Thiết lập các tên, đường link, icon của hệ thống website

## CHƯƠNG 2

### MÔ HÌNH CHỨC NĂNG VÀ MÔ HÌNH DỮ LIỆU BÀI TOÁN QUẢN LÝ TOUR DU LỊCH

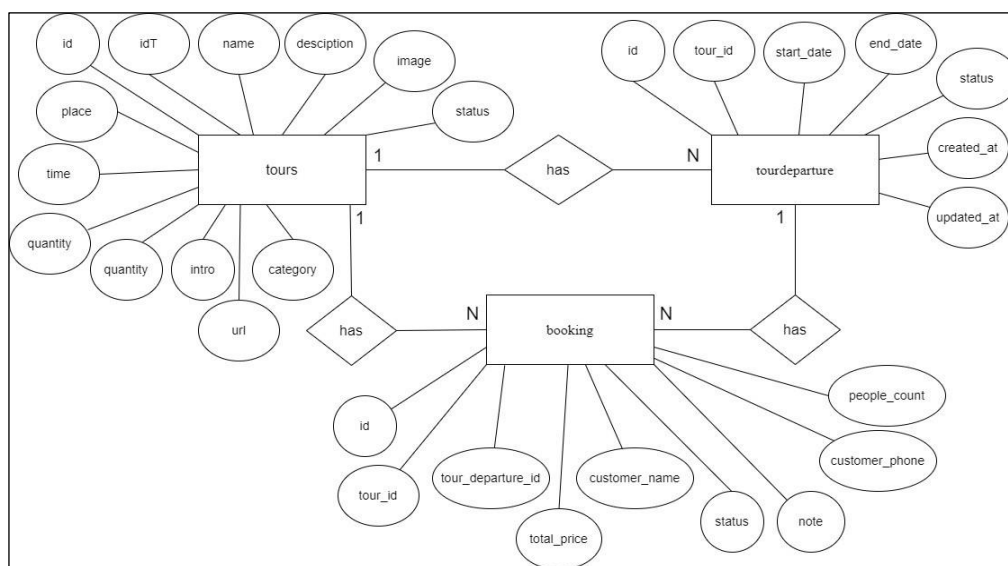
#### 2.1. Mô hình chức năng



Hình 1. Mô hình chức năng của bài toán

#### 2.2. Mô hình cơ sở dữ liệu của bài quản lý tour du lịch

##### 2.2.1. Mô hình thực thể liên hệ



Hình 2. Mô hình thực thể của bài toán

### 2.2.2. Các bảng dữ liệu

- Bảng dữ liệu Tours**

Tên trường	Kiểu dữ liệu	Ý nghĩa
id	BigInt (primary key)	Id tự tăng
idT	Varchar (T + “id”)	Id tour
name	Varchar	Tên của tour
place	Varchar	Điểm đến của tour
price	Int	Giá tour (1 người)
time	Varchar	Thời gian kéo dài của tour
quantity	Int	Số lượng người tối đa của tour
intro	LongText	Giới thiệu tour
image	LongText	Ảnh đại diện đến của tour
status	Int	Trạng thái của tour
category	Int	Thẻ loại tour
url	Text	Đường dẫn của tour

- Bảng dữ liệu Tour\_Departures**

Tên trường	Kiểu dữ liệu	Ý nghĩa
id	Int (primary key)	Id tự tăng
tour_id	Int (foreign key)	Khóa phụ liên kết đến bảng tour
start_date	Date	Ngày bắt đầu
end_date	Date	Ngày kết thúc
status	Int	Trạng thái của lịch trình

- Bảng dữ liệu Bookings**

Tên trường	Kiểu dữ liệu	Ý nghĩa
id	Int (primary key)	Id tự tăng
tour_id	Int (foreign key)	Khóa phụ liên kết đến bảng tour
tour_departure_id	Int (foreign key)	Khóa phụ liên kết đến bảng tour_departure
customer_name	Varchar	Tên khách hàng
customer_email	Varchar	Email khách hàng
customer_phone	Varchar	Số điện thoại của khách hàng



people_count	Int	Số lượng người tham gia trong tour
total_price	Int	Tổng tiền
status	Int	Trạng thái
note	Text	Ghi chú

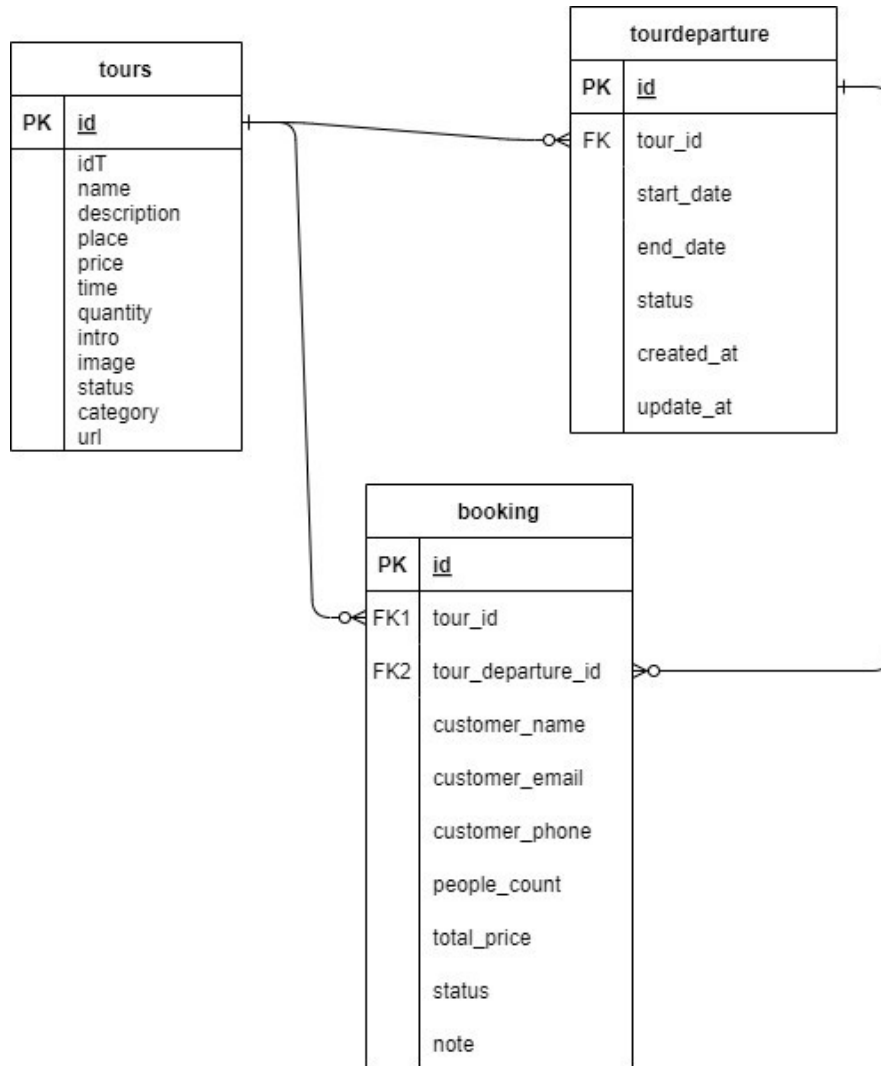
- **Bảng dữ liệu Users**

Tên trường	Kiểu dữ liệu	Ý nghĩa
id	Int (primary key)	Id tự tăng
name	Varchar	Tên tài khoản người dùng
email	Varchar	Email tài khoản người dùng
password	Varchar	Mật khẩu
image	Longtext	Ảnh đại diện của người dùng
phone	Varchar	Số điện thoại của người dùng
gender	Enum	Giới tính của người dùng
address	Varchar	Địa chỉ của người dùng
birthday	date	Ngày sinh của người dùng
role	int	Quyền truy cập hệ thống
status	int	Trạng thái tài khoản người dùng

- **Bảng dữ liệu Menus**

Tên trường	Kiểu dữ liệu	Ý nghĩa
id	Int (primary key)	Id tự tăng
name	Varchar	Tên của menu
link	Varchar	Đường dẫn của menu
icon	Varchar	Icon biểu tượng của menu
status	Int	Trạng thái của menu
note	Text	Ghi chú của menu

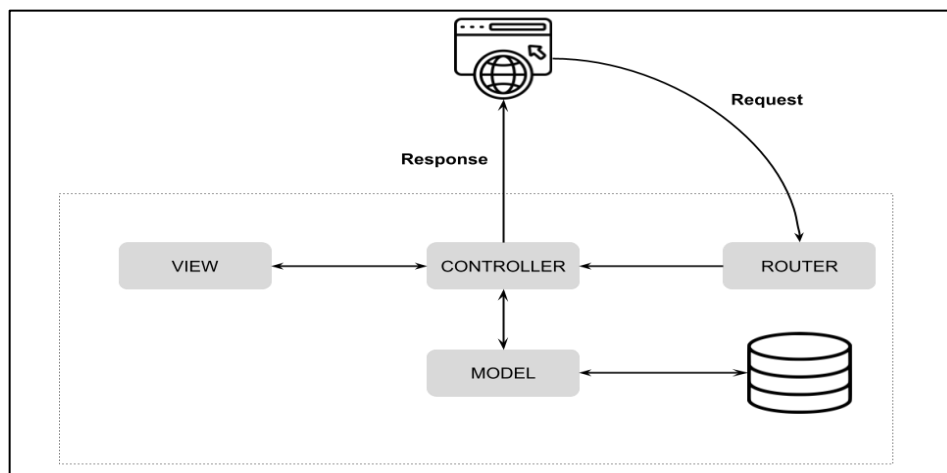
### 2.2.3. Mô hình quan hệ



Hình 3. Mô hình quan hệ

## 2.3. Cấu trúc mô hình MVC trên mã nguồn mở Laravel

### 2.3.1. Cấu trúc chung



Hình 4. Mô hình MVC trên mã nguồn mở Laravel

MVC (Model-View-Controller) là một mô hình kiến trúc phần mềm phổ biến, và Laravel - một framework PHP mã nguồn mở - tuân theo mô hình này.

- **Model:**

Model trong Laravel đại diện cho dữ liệu và các quy tắc nghiệp vụ liên quan đến dữ liệu đó. Model chịu trách nhiệm tương tác với cơ sở dữ liệu, thường sử dụng Eloquent ORM của Laravel để thực hiện các thao tác cơ sở dữ liệu một cách dễ dàng.

- **View:**

View trong Laravel chịu trách nhiệm hiển thị giao diện người dùng. Laravel sử dụng Blade, một engine template mạnh mẽ và dễ sử dụng, để xây dựng các tệp view. Các tệp Blade có phần mở rộng là .blade.php.

- **Controller:**

Controller trong Laravel chịu trách nhiệm xử lý các yêu cầu HTTP, thực hiện logic nghiệp vụ và trả về phản hồi tương ứng. Controller hoạt động như một lớp trung gian giữa Model và View, nhận dữ liệu từ Model và gửi nó đến View.

- **Router:**

Router trong Laravel chịu trách nhiệm định tuyến các yêu cầu HTTP đến các phương thức cụ thể trong Controller. Router đóng vai trò quan trọng trong việc xác định logic điều khiển nào sẽ được thực thi khi một URL cụ thể được truy cập.

### 2.3.2. Cấu hình các route

```
use Illuminate\Support\Facades\Route;

use App\Http\Controllers\Backend\AuthController;
use App\Http\Controllers\BackendUserController;
use App\Http\Controllers\BackendTourController;
use App\Http\Controllers\BackendTourDepartureController;
use App\Http\Controllers\BackendBookingController;
use App\Http\Controllers\BackendMenuController;
use App\Http\Controllers\FrontendAccountController;
```

```

use App\Http\Controllers\Backend\DashboardController;

use App\Http\Middleware\AuthenticateMiddleware;

use App\Http\Middleware>LoginMiddleware;

Route::get('/', function () {
    return view('fontend.home.index');
});

Route::post('booking',
    [BookingController::class, 'create']
    )->name('tour.booking');

Route::group(['prefix'=>'tour'], function(){
    Route::get('index',
        'App\Http\Controllers\Fontend\TourController@index'
    )->name('tour.index');

    Route::get('detail/{url}',
        'App\Http\Controllers\Fontend\TourController@detail'
    )->name('tour.detail');
});

Route::group(['prefix'=>'account'], function(){
    Route::get('login',
        [AccountController::class, 'showLoginForm']
    )->name('account.loginForm');

    Route::get('register',
        [AccountController::class, 'showRegisterForm']
    )->name('account.registerForm');

    Route::post('register',
        [AccountController::class, 'register']
    )->name('account.register');
});

```

```

Route::post('login',
[AccountController::class, 'login']
)->name('account.login');
Route::get('logout',
[AccountController::class, 'logout']
)->name('account.logout');
});
Route::group(['prefix' => 'backend'], function() {
    Route::get('dashboard/index',
[DashboardController::class, 'index']
)->name('dashboard.index')->middleware('admin');
    Route::group(['prefix' => 'auth'], function() {
        Route::get('admin',
[AuthController::class, 'index']
)->name('auth.admin')->middleware('login');
        Route::get('logout',
[AuthController::class, 'logout']
)->name('auth.logout');
        Route::post('login',
[AuthController::class, 'login']
)->name('auth.login');
        Route::get('login',
[AuthController::class, 'login']
)->name('login');
    });
    Route::group(['prefix' => 'user'], function() {
        Route::get('index',

```

```

[UserController::class, 'index']
)->name('user.index')->middleware('admin');
Route::get('create',
[UserController::class, 'formCreate']
)->name('user.formCreate')->middleware('admin');
Route::post('create',
[UserController::class, 'create']
)->name('user.create')->middleware('admin');
Route::get('edit/{id}',
[UserController::class, 'edit']
)->name('user.edit')->middleware('admin');
Route::post('update/{id}',
[UserController::class, 'update']
)->name('user.update')->middleware('admin');
Route::get('delete/{id}',
[UserController::class, 'delete']
)->name('user.delete')->middleware('admin');
});
Route::group(['prefix'=>'tour'], function(){
    Route::get('getTour',
[TourController::class, 'getTour']
)->name('tour.getTour')->middleware('admin');
Route::get('create',
[TourController::class, 'formCreate']
)->name('tour.formCreate')->middleware('admin');
Route::post('create',
[TourController::class, 'create']

```

```

->name('tour.create')->middleware('admin');

Route::get('edit/{id}',
[TourController::class, 'edit']
->name('tour.edit')->middleware('admin');

Route::post('update/{id}',
[TourController::class, 'update']
->name('tour.update')->middleware('admin');


Route::get('delete/{id}',
[TourController::class, 'delete']
->name('tour.delete')->middleware('admin');
});

Route::group(['prefix'=>'tourdeparture'], function(){

Route::get('getTourDeparture',
[TourDepartureController::class, 'getTourDeparture']
->name('tourdeparture.getTourDeparture')->middleware('admin');

Route::get('create',
[TourDepartureController::class, 'formCreate']
->name('tourdeparture.formCreate')->middleware('admin');

Route::post('create',
[TourDepartureController::class, 'create']
->name('tourdeparture.create')->middleware('admin');

Route::get('edit/{id}',
[TourDepartureController::class, 'edit']
->name('tourdeparture.edit')->middleware('admin');

Route::post('update/{id}',
[TourDepartureController::class, 'update']

```

```

    )->name('tourdeparture.update')->middleware('admin');

    Route::get('delete/{id}',
        [TourDepartureController::class, 'delete']
    )->name('tourdeparture.delete')->middleware('admin');
});

Route::group(['prefix'=>'booking'], function(){
    Route::get('getBooking',
        [BookingController::class, 'getBooking']
    )->name('booking.getBooking')->middleware('admin');

    Route::get('edit/{id}',
        [BookingController::class, 'edit']
    )->name('booking.edit')->middleware('admin');

    Route::post('update/{id}',
        [BookingController::class, 'update']
    )->name('booking.update')->middleware('admin');

    Route::get('delete/{id}',
        [BookingController::class, 'delete']
    )->name('booking.delete')->middleware('admin');
});

Route::group(['prefix'=>'menu'], function(){
    Route::get('getMenu',
        [MenuController::class, 'getMenu']
    )->name('menu.getMenu')->middleware('admin');

    Route::get('create',
        [MenuController::class, 'createForm']
    )->name('menu.formCreate')->middleware('admin');

    Route::post('create',

```



```

[MenuController::class, 'create']
)->name('menu.create')->middleware('admin');
Route::get('edit/{id}',
[MenuController::class, 'edit']
)->name('menu.edit')->middleware('admin');
Route::post('update/{id}',
[MenuController::class, 'update']
)->name('menu.update')->middleware('admin');
Route::get('delete/{id}',
[MenuController::class, 'delete']
)->name('menu.delete')->middleware('admin');
});
});

```

### 2.2.3. Cấu hình cơ sở dữ liệu

- **Cấu hình cơ sở dữ liệu User**

```

Schema::create('users', function (Blueprint $table) {
    $table->id();
    $table->string('name');
    $table->string('email')->unique();
    $table->timestamp('email_verified_at')->nullable();
    $table->string('password');
    $table->rememberToken();
    $table->text('image')->nullable();
    $table->string('phone', 20)->nullable();
    $table->enum('gender', ['male', 'female', 'other'])->nullable();
    $table->text('address')->nullable();
    $table->dateTime('birthday')->nullable();

```

```

$stable->bigInteger('role')->default(2);

$stable->boolean('status')->default(true);

$stable->timestamps();

});

```

- **Cấu hình cơ sở dữ liệu Tours**

```

Schema::create('Tours', function (Blueprint $table) {
    $table->id();

    $table->string('idT')->nullable(); // Thêm trường idT

    $table->string('name');

    $table->longtext('description')->nullable(); // Thêm trường description

    $table->string('place')->nullable();

    $table->integer('price')->nullable();

    $table->string('time')->nullable();

    $table->integer('quantity')->nullable();

    $table->longtext('intro')->nullable();

    $table->longtext('image')->nullable();


    $table->boolean('status')->default(true);

    $table->boolean('category')->default(true);

    $table->text('url')->nullable();

    $table->timestamps();

});

```

- **Cấu hình cơ sở dữ liệu TourDepartures**

```

Schema::create('tour_departures', function (Blueprint $table) {
    $table->id();

    $table->unsignedBigInteger('tour_id');

    $table->dateTime('start_date');

```

```

$table->dateTime('end_date')->nullable();

$table->integer('status');

$table->integer('price')->nullable();

$table->timestamps();

});

```

- **Cấu hình cơ sở dữ liệu Bookings**

```

Schema::create('bookings', function (Blueprint $table) {
    $table->id();

    $table->unsignedBigInteger('tour_id');
    $table->unsignedBigInteger('tour_departure_id');
    $table->string('customer_name');
    $table->string('customer_email')->nullable();
    $table->string('customer_phone');
    $table->integer('people_count');
    $table->integer('total_price');
    $table->string('note')->nullable();
    $table->tinyInteger('status')->default(0);
    $table->tinyInteger('status')->default(0);
    $table->timestamps();

    // chỉ mục tăng hiệu xuất truy vấn trong csdl
    $table->index('tour_id');
    $table->index('tour_departure_id');

});

```

- **Cấu hình cơ sở dữ liệu Menus**

```

Schema::create('menus', function (Blueprint $table) {
    $table->id();

    $table->string('name');

```

```

    $table->string('link');

    $table->string('icon')->nullable();

    $table->text('note')->nullable();

    $table->boolean('status')->default(true);

    $table->timestamps();

    });

```

#### 2.2.4. Tầng Controller

- **AuthController**

```

namespace App\Http\Controllers\Backend;

use Illuminate\Http\Request;

use App\Http\Controllers\Controller;

use App\Http\Requests\AuthRequest;

use Illuminate\Support\Facades\Auth;

class AuthController extends Controller
{
    public function index()
    {
        if(Auth::id()>0){
            return redirect()->route('dashboard.index');
        }

        return view('backend.auth.login');
    }

    public function login(AuthRequest $request){
        $credentials = [
            'email' => $request->input('email'),

```

```

        'password' => $request->input('password')
    ];
    if(Auth::attempt($credentials))
    {
        $user = Auth::user();
        return redirect()->route('dashboard.index',['user' => $user]
        )->with('success','Đăng nhập thành công');
    }
    return redirect()->route('auth.admin')->with('error','Email hoặc mật khẩu
không chính xác!');
}

public function logout(Request $request){
    Auth::logout();
    $request->session()->invalidate();
    $request->session()->regenerateToken();
    return redirect()->route('auth.admin');
}
}

```

- **UserController**

```

namespace App\Http\Controllers\Backend;
use App\Models\User;
use Illuminate\Support\Facades\Hash;
use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use Carbon\Carbon;

class UserController extends Controller

```

```

{
    public function index(Request $request){
        $search = $request->query('keyword');
        if ($search) {
            $users = User::where('name', 'like', '%' . $search . '%')
                ->orWhere('email', 'like', '%' . $search . '%')
                ->orWhere('address', 'like', '%' . $search . '%')
                ->orWhere('phone', 'like', '%' . $search . '%')
                ->get();
        } else {
            $users = User::orderBy('id', 'desc')->get();
        }

        return view('backend.user.index', ['users'=>$users, 'title'=>'Danh sách thành
viên']);
    }

    public function formCreate(){
        return view('backend.user.create', ['title'=>'Thêm mới thành viên']);
    }

    function create(Request $request){
        $user = new User;
        $user->name = $request->name;
        $user->email = $request->email;
        $user->phone = $request->phone;
        $user->address = $request->address;
        $user->birthday = $request->birthday;
        $user->gender = $request->gender;
        $user->role = $request->role;
    }
}

```

```

$user->status = $request->status;

// Mã hóa mật khẩu trước khi lưu

$user->password = Hash::make($request->password);

if (isset($_FILES['image']) && $_FILES['image']['error'] ===
UPLOAD_ERR_OK) {

    $image = 'data:image/png;base64,' .
base64_encode(file_get_contents($_FILES['image']['tmp_name']));

    $user->image = $image;

}

$user->save();

return redirect()->route('user.index')->with('success', "Thêm thành công!");

}

public function edit($id){

    $user = User::where('id',$id)->first();

    // Chuyển đổi 'birthday' thành đối tượng Carbon nếu không phải là null

    if ($user->birthday) {

        $user->birthday = Carbon::parse($user->birthday);

    }

    return view('backend.user.edit',['user'=>$user,'title'=>'Chỉnh sửa thông
thành viên']);

}

function update(Request $request,$id){

    $user = User::findOrFail($id);

    $user->name = $request->name;

    $user->email = $request->email;

```

```

$user->phone = $request->phone;

$user->address = $request->address;

$user->birthday = $request->birthday;

$user->gender = $request->gender;

$user->role = $request->role;

$user->status = $request->status;

// Mã hóa mật khẩu trước khi lưu

$user->password = Hash::make($request->password);

if (isset($_FILES['image']) && $_FILES['image']['error'] ===
UPLOAD_ERR_OK) {

    $image = 'data:image/png;base64,' .
base64_encode(file_get_contents($_FILES['image']['tmp_name']));

    $user->image = $image;

}

$user->save();

return redirect()->route('user.index')->with('success', "Chỉnh sửa thành
công!");

}

function delete($id){

    $user = User::where('id', $id)->first();

    $user->delete();

    return redirect()->route('user.index')->with('success','Xóa thành công');

}

}

```

- **TourController**

```

namespace App\Http\Controllers\Backend;

use App\Models\TourModel;

```



```

use App\Http\Controllers\Controller;

use Illuminate\Http\Request;

use Illuminate\Support\Str;

class TourController extends Controller
{
    public function getTour(Request $request)
    {
        $search = $request->query('keyword');

        if ($search) {
            $tours = TourModel::where('name', 'like', '%' . $search . '%')
                ->orWhere('description', 'like', '%' . $search . '%')
                ->orWhere('place', 'like', '%' . $search . '%')
                ->orWhere('price', 'like', '%' . $search . '%')
                ->orWhere('time', 'like', '%' . $search . '%')
                ->get();
        } else {
            $tours = TourModel::orderBy('id', 'desc')->get();
        }

        return view('backend.tour.getTour', ['tours'=>$tours, 'title'=>'Danh sách tour
du lịch']);
    }

    public function formCreate()
    {
        return view('backend.tour.create', ['title'=>'Thêm tour du lịch']);
    }
}

```

```

public function create(Request $request)
{
    $tour = new TourModel;

    $tour->name = $request->name;

    $tour->description = $request->description;

    $tour->place = $request->place;

    $tour->price = $request->price;

    $tour->time = $request->time;

    $tour->quantity = $request->quantity;

    $tour->intro = $request->intro;

    $tour->category = $request->category;

    $tour->status = $request->status;

    // Tạo giá trị cho trường idT và url

    $tour->idT = 'T' . now()->format('YmdHis'); // Format ngày tháng năm hiện
    tại

    $tour->url = Str::slug($request->name); // Tạo URL từ tên tour

    if (isset($_FILES['image']) && $_FILES['image']['error'] ===
    UPLOAD_ERR_OK) {

        $image = 'data:image/png;base64,' .
        base64_encode(file_get_contents($_FILES['image']['tmp_name']));

        $tour->image = $image;

    }

    $tour->save();

    return redirect()->route('tour.getTour')->with('success', "Thêm thành
    công!");

}

public function edit($id)

```

```

{
    $tour = TourModel::where('id',$id)->first();
    return view('backend.tour.edit',['tour'=>$tour,'title'=>'Chỉnh sửa tour du
    lịch']);
}

public function update(Request $request,$id)
{
    $tour = TourModel::findOrFail($id);
    $tour->name = $request->name;
    $tour->description = $request->description;
    $tour->place = $request->place;
    $tour->price = $request->price;
    $tour->time = $request->time;
    $tour->quantity = $request->quantity;
    $tour->intro = $request->intro;
    $tour->status = $request->status;
    $tour->category = $request->category;
    $tour->url = Str::slug($request->name); // Tạo URL từ tên tou
    if (isset($_FILES['image']) && $_FILES['image']['error'] ===
    UPLOAD_ERR_OK) {
        $image = 'data:image/png;base64,' .
        base64_encode(file_get_contents($_FILES['image']['tmp_name']));
        $tour->image = $image;
    }
    $tour->save();
    return redirect()->route('tour.getTour')->with('success', "Chỉnh sửa thành
    công!");
}

```

```

public function delete($id)
{
    $tour = TourModel::where('id', $id)->first();
    $tour->delete();
    return redirect()->route('tour.getTour')->with('success','Xóa thành công');
}
}

```

- **TourDepartureController**

```

namespace App\Http\Controllers\Backend;
use App\Models\TourDepartureModel;
use App\Models\TourModel;
use Carbon\Carbon;
use App\Http\Controllers\Controller;
use Illuminate\Http\Request;

class TourDepartureController extends Controller
{
    public function getTourDeparture(Request $request)
    {
        $search = $request->query('keyword');

        if ($search) {
            $tour_departures = TourDepartureModel::whereHas('tour', function
($query) use ($search) {
                $query->where('name', 'like', '%' . $search . '%')->orWhere('idT',
'like', '%' . $search . '%');
            }->orWhere('tour_id', 'like', '%' . $search . '%')->get();

```

```

    } else {
        $tour_departures = TourDepartureModel::orderBy('id', 'desc')->get();
    }

    return view('backend.tourdeparture.getTourDeparture', ['tour_departures'
=> $tour_departures, 'title' => 'Danh sách lịch khởi hành']);
}

public function formCreate(Request $request)
{
    $tours = TourModel::all(); // Lấy danh sách các tour

    return view('backend.tourdeparture.create', ['title' => 'Thêm lịch khởi hành',
'tours' => $tours]);
}

public function create(Request $request)
{
    $tourdeparture = new TourDepartureModel();
    $tourdeparture->tour_id = $request->tour_id;
    $tourdeparture->start_date = $request->start_date;
    $tourdeparture->end_date = $request->end_date;
    $tourdeparture->status = $request->status;
    $tourdeparture->save();

    return redirect()->route('tourdeparture.getTourDeparture')->with('success',
"Thêm thành công!");
}

public function edit($id)
{
    $tourdeparture = TourDepartureModel::find($id); // Sử dụng find để tìm bản
ghi với id

```

```

$tours = TourModel::all(); // Lấy danh sách các tour

if ($tourdeparture->start_date) {
    $tourdeparture->start_date = Carbon::parse($tourdeparture->start_date);
}

if ($tourdeparture->end_date) {
    $tourdeparture->end_date = Carbon::parse($tourdeparture->end_date);
}

return view('backend.tourdeparture.edit', ['tourdeparture' => $tourdeparture,
'tours' => $tours, 'title' => 'Chỉnh sửa lịch trình']);
}

public function update(Request $request, $id)
{
    $tourdeparture = TourDepartureModel::findOrFail($id);
    $tourdeparture->tour_id = $request->tour_id;
    $tourdeparture->start_date = $request->start_date;
    $tourdeparture->end_date = $request->end_date;
    $tourdeparture->status = $request->status;
    $tourdeparture->save();

    return redirect()->route('tourdeparture.getTourDeparture')->with('success',
'Sửa thành công!');
}

public function delete($id)
{
    $tour = TourDepartureModel::where('id', $id)->first();
    $tour->delete();

    return redirect()->route('tourdeparture.getTourDeparture')->with('success', 'Xóa thành công');
}

```

```
}  
}
```

- **BookingController**

```
namespace App\Http\Controllers\Backend;  
use App\Http\Controllers\Controller;  
use Illuminate\Http\Request;  
use App\Models\BookingModel;  
class BookingController extends Controller  
{  
    public function getBooking(Request $request)  
    {  
        // Bắt đầu query để lấy tất cả booking  
        $query = BookingModel::query();  
  
        // Lấy giá trị status từ request và thêm điều kiện nếu có  
        $status = $request->query('status');  
        if ($status !== null && $status !== "") {  
            $query->where('status', $status);  
        }  
  
        // Lấy giá trị keyword từ request và thêm điều kiện nếu có  
        $search = $request->query('keyword');  
        if ($search) {  
            $query->where(function ($q) use ($search) {  
                $q->whereHas('tour', function ($q2) use ($search) {  
                    $q2->where('name', 'like', '%' . $search . '%')  
                    ->orWhere('idT', 'like', '%' . $search . '%');  
                });  
            });  
        }  
    }  
}
```

```

        })->orWhere('customer_name', 'like', '%' . $search . '%')
        ->orWhere('customer_phone', 'like', '%' . $search . '%')
        ->orWhere('customer_email', 'like', '%' . $search . '%');
    });
}

// Lấy danh sách booking với các điều kiện đã xây dựng
$bookings = $query->orderBy('id', 'desc')->get();

return view('backend.booking.getBooking', ['bookings' => $bookings, 'title' =>
'Danh sách đặt tour']);
}

public function create(Request $request)
{
    $booking = new BookingModel;
    $booking->tour_id = $request->tour_id;
    $booking->tour_departure_id = $request->tour_departure_id;
    $booking->customer_name = $request->customer_name;
    $booking->customer_email = $request->customer_email;
    $booking->customer_phone = $request->customer_phone;
    $booking->people_count = $request->people_count;
    $booking->total_price = $request->total_price;
    $booking->note = $request->note;
    $booking->save();

    return redirect()->back()->with('success', 'Bạn đã đặt Tour thành công!');
}

public function delete($id)
{

```



```

        $booking = BookingModel::where('id', $id)->first();

        $booking->delete();

        return redirect()->route('booking.getBooking')->with('success','Xóa thành
công');
    }

```

```

    public function edit($id)
    {
        $booking = BookingModel::where('id',$id)->first();

        return view('backend.booking.edit',['booking'=>$booking,'title'=>'Chỉnh
sửa trạng thái booking']);
    }

```

```

    public function update(Request $request, $id)
    {
        $booking = BookingModel::findOrFail($id);

        $booking->status = $request->status;

        $booking->save();

        return redirect('backend/booking/getBooking')->with('success', "Sửa thành
công!");
    }
}

```

- **DashboardController**

```

namespace App\Http\Controllers\Backend;

use App\Http\Controllers\Controller;

class DashboardController extends Controller
{
    public function index(){

```

```

        return view('backend.dashboard.index');
    }
}

```

- **MenuController**

```

namespace App\Http\Controllers\Backend;
use App\Models\MenuModel;
use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
class MenuController extends Controller
{
    public function getMenu()
    {
        $menus = MenuModel::where('status', true)->get();
        $title= "Danh sách menu";
        return view('backend.menu.getMenu', compact('menus','title'));
    }
    public function createForm()
    {
        $title= "Thêm mới menu";
        return view('backend.menu.create',compact('title'));
    }
    public function create(Request $request)
    {
        $menu = new MenuModel();
        $menu->name = $request->name;
        $menu->link = $request->link;
        $menu->icon = $request->icon;
    }
}

```

```

$menu->status = $request->status;

$menu->note = $request->note;

$menu->save();

return redirect()->route('menu.getMenu')->with('success', "Thêm thành
công!");
}

public function edit($id)
{
    $title= "Chỉnh sửa thông tin menu";

    $menu = MenuModel::findOrFail($id);

    return view('backend.menu.edit', compact('menu','title'));
}

public function update(Request $request, $id)
{
    $menu = MenuModel::findOrFail($id);

    $menu->name = $request->name;

    $menu->link = $request->link;

    $menu->icon = $request->icon;

    $menu->status = $request->status;

    $menu->note = $request->note;

    $menu->save();

    return redirect()->route('menu.getMenu')->with('success', 'Sửa menu thành
công.');
```

```

}

public function destroy($id)
{
    $menu = MenuModel::findOrFail($id);

```

```

$menu->delete();

return redirect()->route('menu.getMenu')->with('success', 'Xóa menu thành
công.');
```

```

    }
}

```

### 2.2.5. Tầng Model

- **Model User**

```

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Foundation\Auth\User as Authenticatable;
use Illuminate\Notifications\Notifiable;
use Laravel\Sanctum\HasApiTokens;

class User extends Authenticatable
{
    use HasApiTokens, HasFactory, Notifiable;

    protected $table = "users";

    protected $dates = ['birthday'];

    protected $fillable = [
        'name',
        'email',
        'password',
        'phone',
        'address',
        'birthday',
        'image',
        'gender',
    ];
}

```

```

        'role',

        'status',
    ];

    protected $hidden = [

        'password',

        'remember_token',
    ];

    protected $casts = [

        'email_verified_at' => 'datetime',

        'password' => 'hashed',
    ];

}

```

- **Model Tour**

```

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class TourModel extends Model
{
    use HasFactory;

    protected $table = 'Tours';

    public function tourDepartures()
    {
        return $this->hasMany(TourDepartureModel::class, 'tour_id', 'id');
    }
}

```

```

public function bookings()
{
    return $this->hasMany(BookingModel::class, 'tour_id', 'id');
}
}

```

- **Model TourDeparture**

```

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class TourDepartureModel extends Model
{
    use HasFactory;

    protected $table = 'tour_departures';

    public function tour()
    {
        return $this->belongsTo(TourModel::class, 'tour_id', 'id');
    }

    public function bookings()
    {
        return $this->hasMany(BookingModel::class, 'tour_departure_id', 'id');
    }
}

```

- **Model Booking**

```

namespace App\Models;

```

```

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
class BookingModel extends Model
{
    use HasFactory;
    protected $table = "bookings";
    public function tour()
    {
        return $this->belongsTo(TourModel::class, 'tour_id', 'id');
    }
    public function tourDeparture()
    {
        return $this->belongsTo(TourDepartureModel::class, 'tour_departure_id',
'id');
    }
}

```

- **Model Menu**

```

namespace App\Models;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
class MenuModel extends Model
{
    use HasFactory;
    protected $table = "menus";
}

```

### 2.2.6. Tầng View

Do số lượng quá lớn chúng em xin trình bày một tầng view của Tour

- **getTour.blade.php**

```
@extends('backend.layout.layout')

@section('content')

    @include('backend.layout.component.breadcrumb')

    <div class="wrapper wrapper-content fadeInRight">

        <div class="row">

            <div class="col-lg-12">

                <div class="ibox float-e-margins">

                    @include('backend.layout.component.toolbox')

                    <div class="ibox-content">

                        @include('backend.tour.component.filter')

                        @include('backend.tour.component.table')

                    </div>

                </div>

            </div>

        </div>

    </div>

@endsection
```

- **filter.blade.php (backend.tour.component.filter)**

```
<form action="{ { route('tour.getTour') } }" method="GET">

    <div class="filter-wrapper">

        <div class="uk-flex uk-flex-middle uk-flex-space-between">

            <div class="input-group">

                <input

                    type="text"

                    name = "keyword"

                >

            </div>

        </div>

    </div>

</form>
```



```

        value="{{ request('keyword') ?: old('keyword') }}"
        placeholder="Nhập từ khóa....."
        class="form-control" >
    <span class="input-group-btn">
        <button type="submit" name="search" value="search" class="btn
btn-success mb0 btn-xl mr10">
            Tìm kiếm
        </button>
    </span>
</div>
<div class="action text-right">
    <div class="uk-flex uk-flex-middle">
        <a href="{{ route('tour.formCreate') }}" class="btn btn-danger"><i
class="fa fa-plus mr5"></i>Thêm tour</a>
    </div>
</div>
</div>
</div>
</form>

```

- **table.blade.php (backend.tour.component.table)**

```

<table class="table table-bordered">
    <thead>
        <tr>
            <th>Mã Số Tour</th>
            <th>Tên</th>
            <th>Địa điểm</th>
            <th>Giá</th>

```

```

        <th class="text-center">Thời gian</th>
        <th class="text-center">Trạng thái</th>
        <th class="text-center">Thao tác</th>
    </tr>
</thead>
<tbody>
    @if(isset($tours)&& is_object($tours))
    @foreach($tours as $tour)
    <tr>
        <td> {{ $tour->idT }}</td>
        <td> {{ $tour->name }}</td>
        <td> {{ $tour->place }}</td>
        <td> {{ $tour->price }}</td>
        <td> {{ $tour->time }}</td>
        <td class="text-center">
            @if($tour->status == 1)
                <span class="label label-primary">Đang hoạt động</span>
            @else
                <span class="label label-danger">Ngưng hoạt động</span>
            @endif
        </td>
        <td class="text-center table-btn ">
            <a href="{{ route('tour.edit', $tour->id) }}" class="btn btn-success"><i
class="fa fa-edit"></i></a>
            <a href="{{ route('tour.delete', $tour->id) }}" class="btn btn-danger"><i
class="fa fa-trash"></i></a>
        </td>
    </tr>
    @endforeach
</tbody>
</table>

```

```

</tr>

@endforeach

@endif

</tbody>

</table>

```

- **Thêm tour - Create.blade.php**

Bởi vì form thêm tour và chỉnh sửa tour giống nhau chỉ khác mỗi route nên chúng em xin phép chỉ trình bày một form

```

@extends('backend.layout.layout')

@section('content')

    @include('backend.layout.component.breadcrumb')

    <div class="wrapper wrapper-content fadeInRight">

        <form action="{ { route('tour.create') } }" method="post" class="box"
        enctype="multipart/form-data">

            @csrf

            <div class="wrapper wrapper-content animated fadeInRight">

                <div class="text-right mb15">

                    <button type="submit" class="btn btn-info "><i class="fa fa-plus
                    mr5"></i>Lưu lại</button>

                </div>

                <div class="row">

                    <div class="col-lg-4">

                        <div class="panel-title">Thông tin chung</div>

                        <div class="panel-description"><p>Nhập thông tin của người sử
                        dụng</p>

                        <p>Lưu ý: Những trường đánh dấu <span class="text-
                        danger">(*)</span> là bắt buộc</p>

                    </div>

```

```

</div>

<div class="col-lg-8">
  <div class="ibox">
    <div class="ibox-content">
      <div class="row mb15">
        <div class="col-lg-12">
          <div class="form-row">
            <label for="" class="control-label text-right">Tên
tour du lịch

            <span class="text-danger">(*)</span>
          </label>
          <input
            type="text"
            class="form-control"
            name="name"
            value="{{ old('name') }}"
            autocomplete="off"
          >
        </div>
      </div>
    </div>
  </div>
</div>

<div class="row mb15">
  <div class="col-lg-6">
    <div class="form-row">
      <label for="" class="control-label text-right">Địa
điểm

      <span class="text-danger">(*)</span>
    </div>
  </div>
</div>

```

```

</label>

<input
  type="text"
  class="form-control"
  name="place"
  value="{{ old('place') }}"
  autocomplete="off"
>
</div>
</div>
<div class="col-lg-6">
  <label for="" class="control-label text-right">Loại tour
    <span class="text-danger">(*)</span>
  </label>
  <select name="category" id="" class="form-control">
    <option value="1">Trong nước</option>
    <option value="0">Quốc tế</option>
  </select>
</div>
</div>
<div class="row mb15">
  <div class="col-lg-6">
    <div class="form-row">
      <label for="" class="control-label text-left">Giá ( 1
        <span class="text-danger">(*)</span>
      </label>

```

thành viên )

```

        <input
        type="text"
        class="form-control"
        name="price"
        value="{{ old('price') }}"
        autocomplete="off"
        >
    </div>
</div>
<div class="col-lg-6">
    <div class="form-row">
        <label for="" class="control-label text-left">Số lượng
        người cho phép

        <span class="text-danger">(*)</span>
    </label>
    <input type="text"
    class="form-control"
    value="{{ old('quantity') }}"
    name="quantity">
    </div>
</div>
</div>
<div class="row mb15">
    <div class="col-lg-6">
        <div class="form-row">
            <label for="" class="control-label text-left">Thời
            gian

```

```

        <span class="text-danger">(*)</span>
    </label>

    <input type="text"
    class="form-control"
    name="time"
    value="{{ old('time') }}" >

</div>
</div>
<div class="col-lg-6">
    <div class="form-row">
        <label for="" class="control-label text-left">Trạng
thái</label>

        <select name="status" id="" class="form-control">
            <option value="1">Hoạt động</option>
            <option value="0">Ngưng hoạt động</option>
        </select>
    </div>
</div>
</div>
<div class="row mb15">
    <div class="col-lg-12">
        <div class="form-row">
            <label for="" class="control-label text-left">Hình
ảnh

            <span class="text-danger">(*)</span>
        </label>

```

```

        <input type="file" class="form-control"
name="image">

    </div>

</div>

</div>

<div class="row mb15">
    <div class="col-lg-12">
        <div class="form-row">
            <label for="" class="control-label text-left"> Mô tả
                <span class="text-danger">(*)</span>
            </label>
            <textarea
                class="inbox-textarea form-control"
name="description">{{ old('description') }}</textarea>
        </div>
    </div>
</div>

<div class="row mb15">
    <div class="col-lg-12">
        <div class="form-row">
            <label for="" class="control-label text-left"> Giới
thiệu
                <span class="text-danger">(*)</span>
            </label>
            <textarea
                class="summernote inbox-textarea form-control
mb15"

```



name="intro"

style="display: none;">{{old('intro')}}</textarea>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</form>

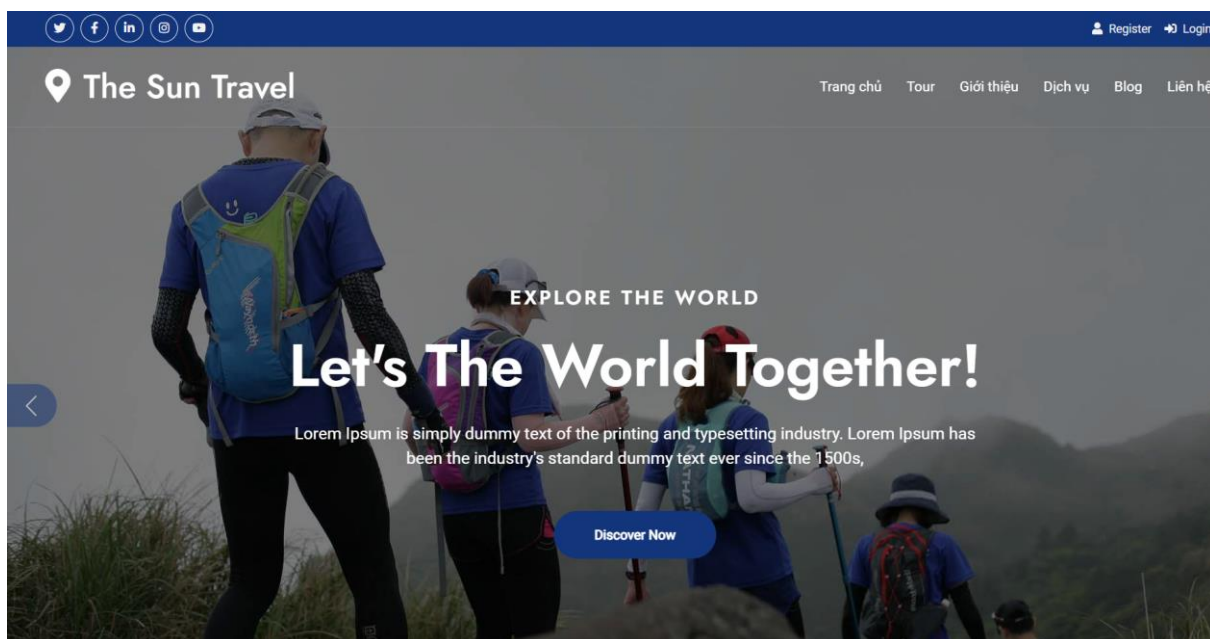
</div>

@endsection

## CHƯƠNG 3

### CÀI ĐẶT VÀ CÁC GIAO DIỆN WEBSITE

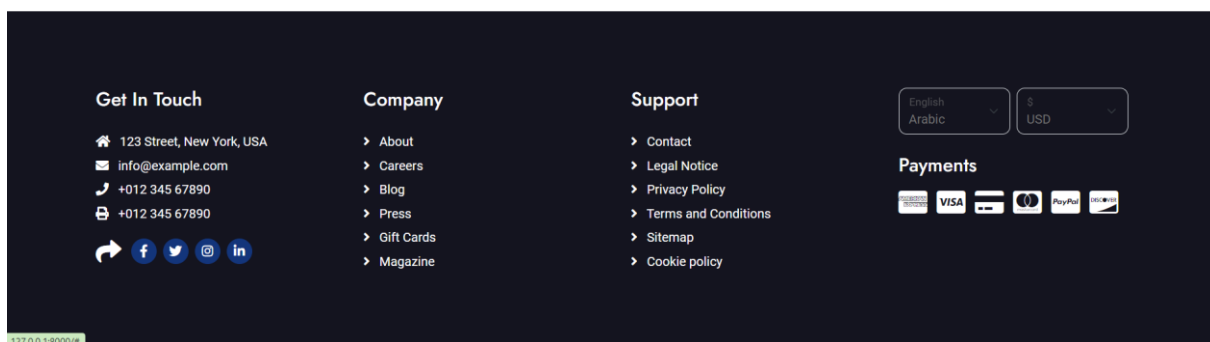
#### 3.1. Giao diện trang chủ



Hình 5. Giao diện menu và banner của trang chủ



Hình 6. Giao diện menu và banner được kế thừa ở mọi trang



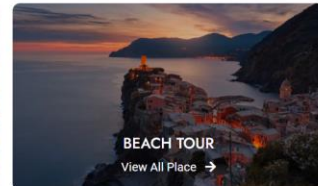
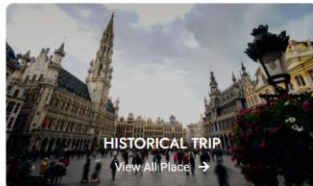
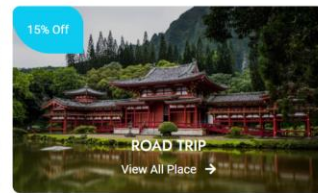
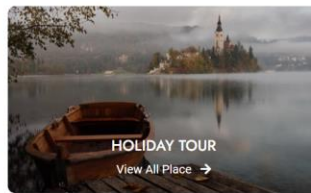
Hình 7. Giao diện Footer được kế thừa ở mọi trang

## The World

Lorem ipsum dolor sit amet consectetur adipisicing elit. Laborum tempore nam, architecto doloremque velit explicabo? Voluptate sunt eveniet fuga eligendi Expedita laudantium fugiat corrupti eum cum repellat a laborum quasi.

National Tour Category

International tour Category



TRAVEL GUIDE

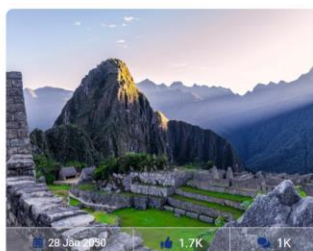
## Meet Our Guide



OUR BLOG

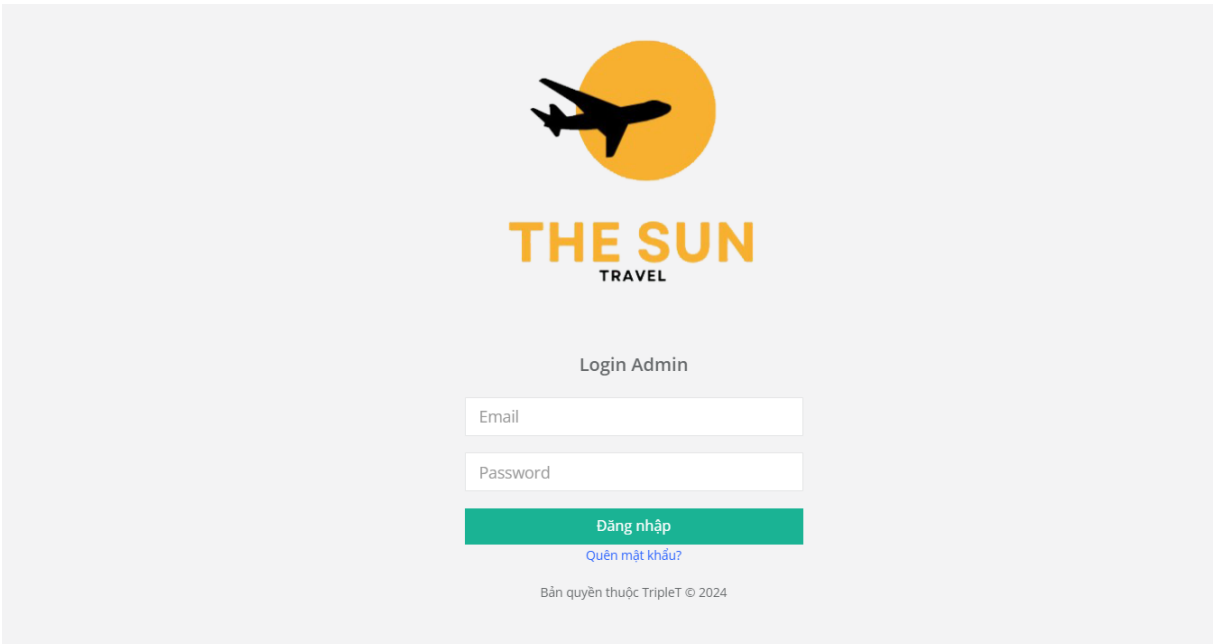
## Popular Travel Blogs

Lorem ipsum dolor sit amet consectetur adipisicing elit. Deleniti deserunt tenetur sapiente atque. Magni non explicabo beatae sit, vel reiciendis consectetur numquam id similique sunt error obcaecati ducimus officia maiores.

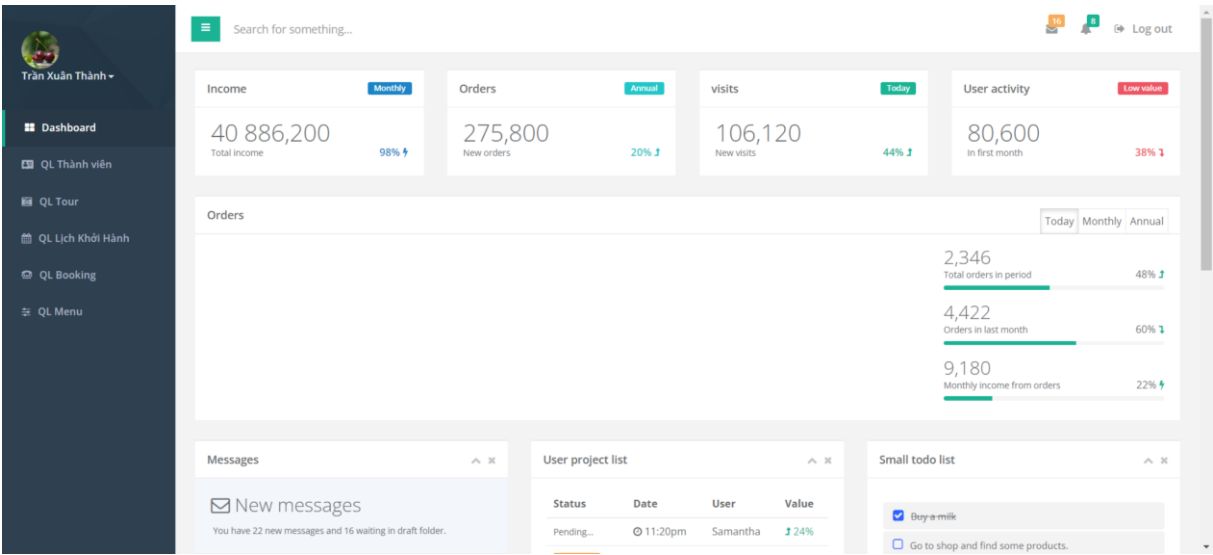


Hình 9. Giao diện thân của trang chủ

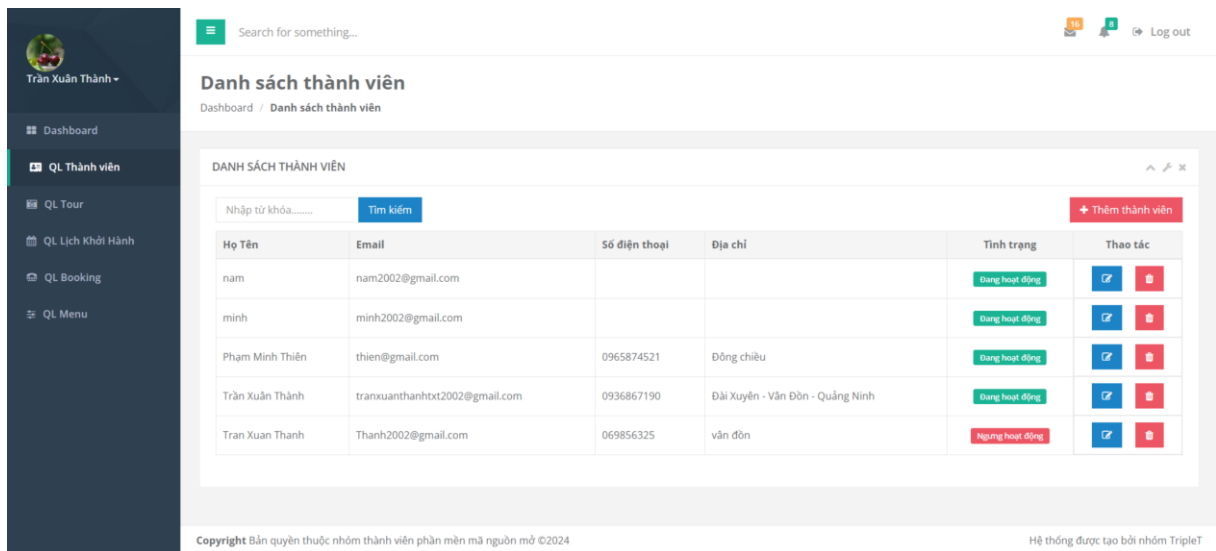
3.2. Giao diện các trang cho nhà quản trị



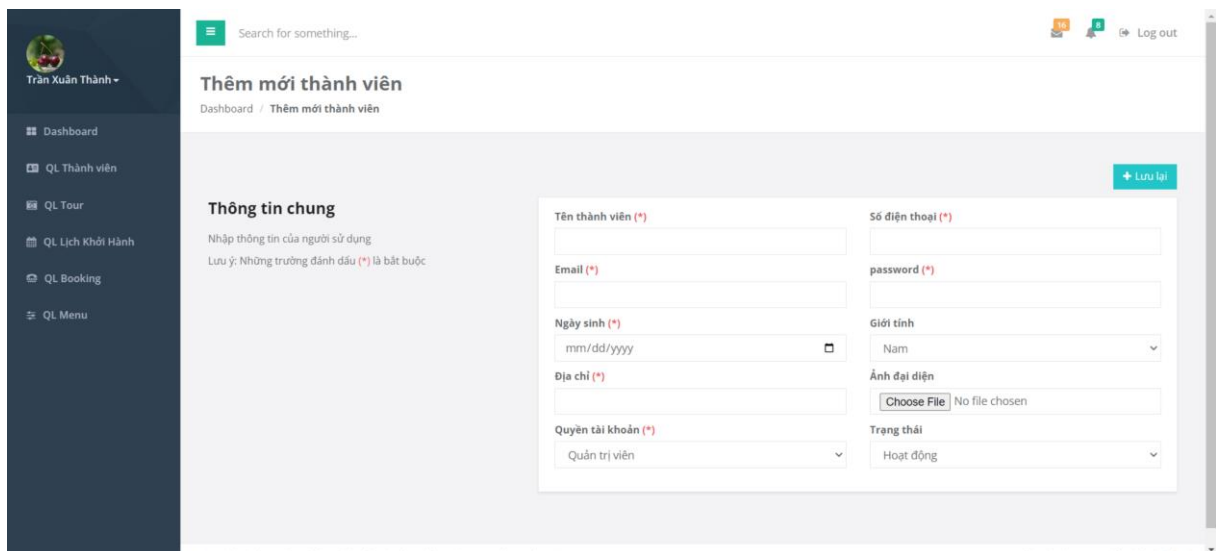
Hình 10. Giao diện trang đăng nhập vào hệ thống quản trị



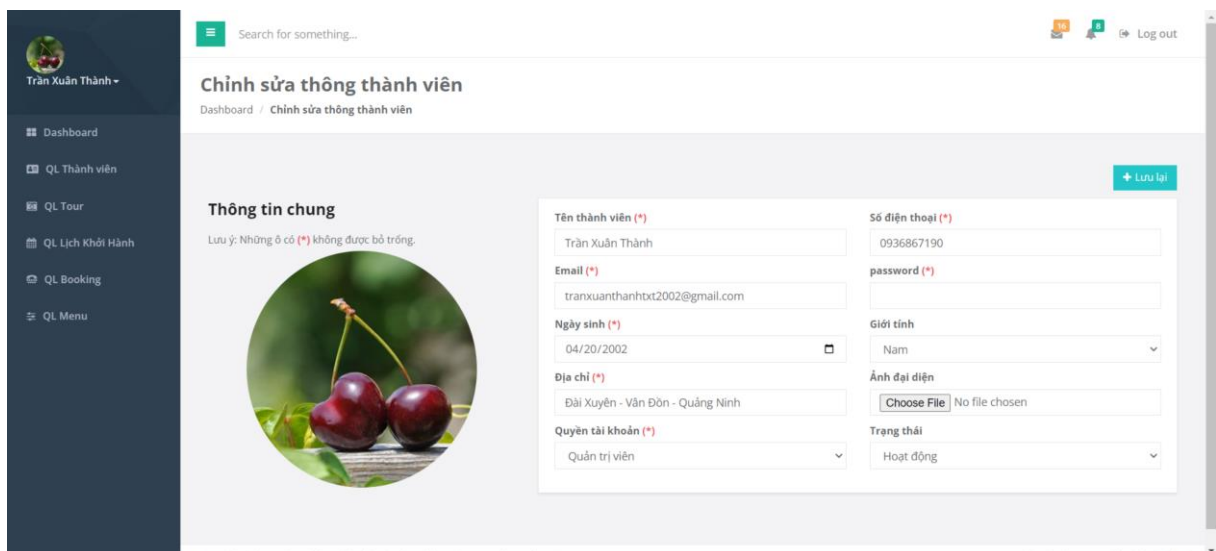
Hình 11. Giao diện trang dashboard



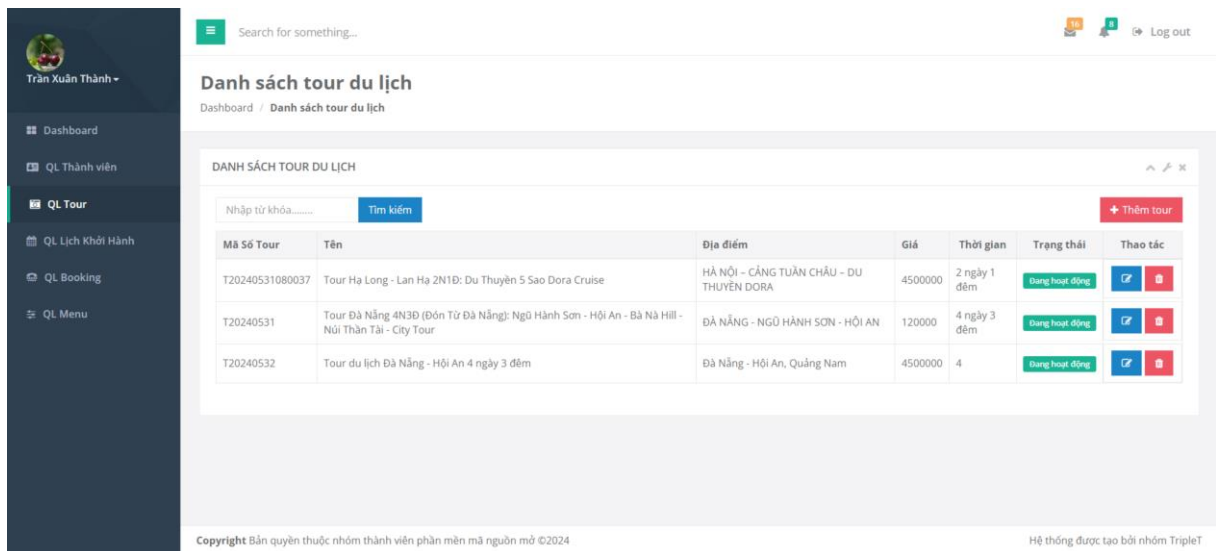
Hình 12. Giao diện danh sách thành viên-user



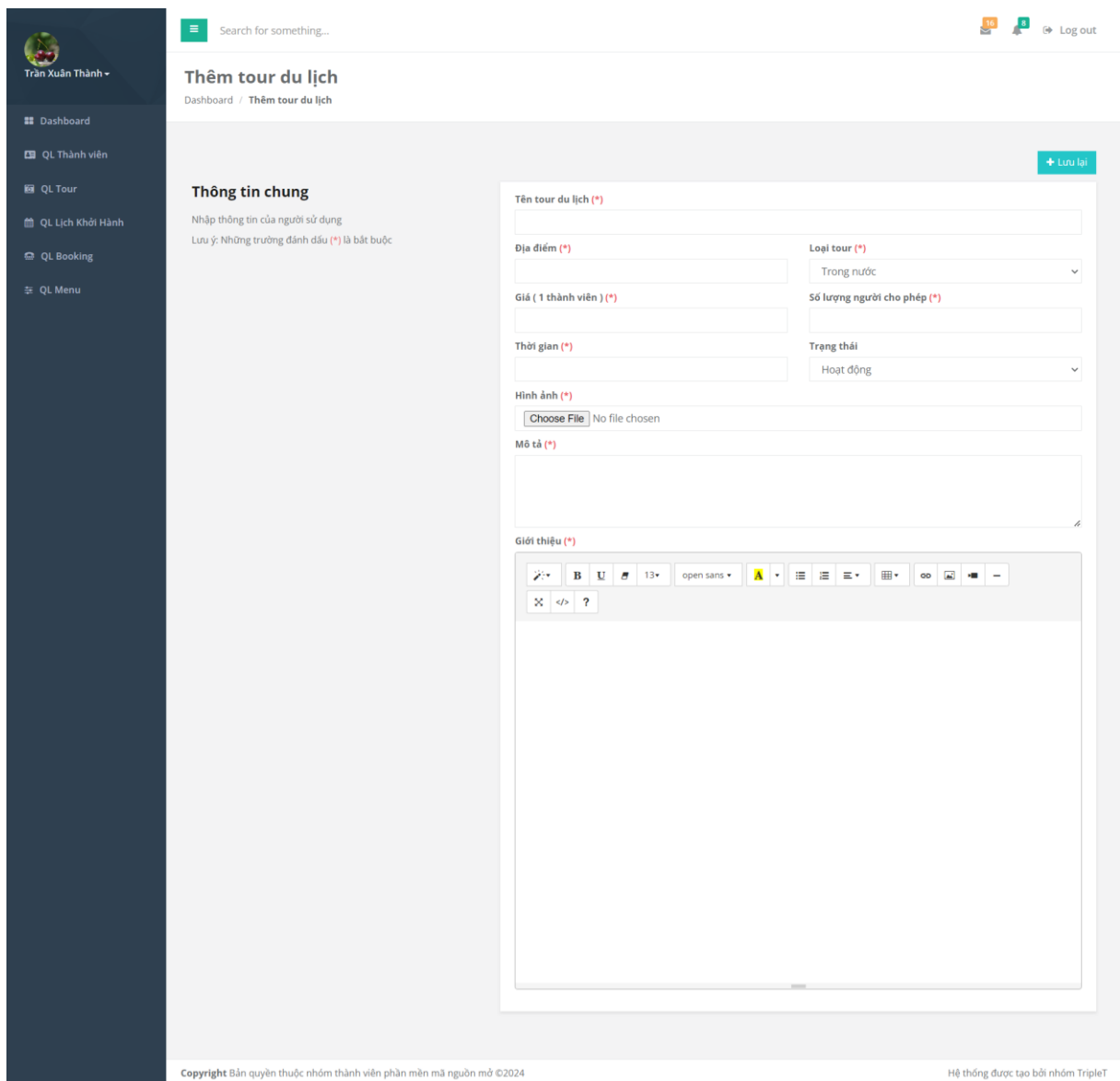
Hình 13. Giao diện danh thêm thành viên-user



Hình 14. Giao diện danh chỉnh sửa thành viên-user

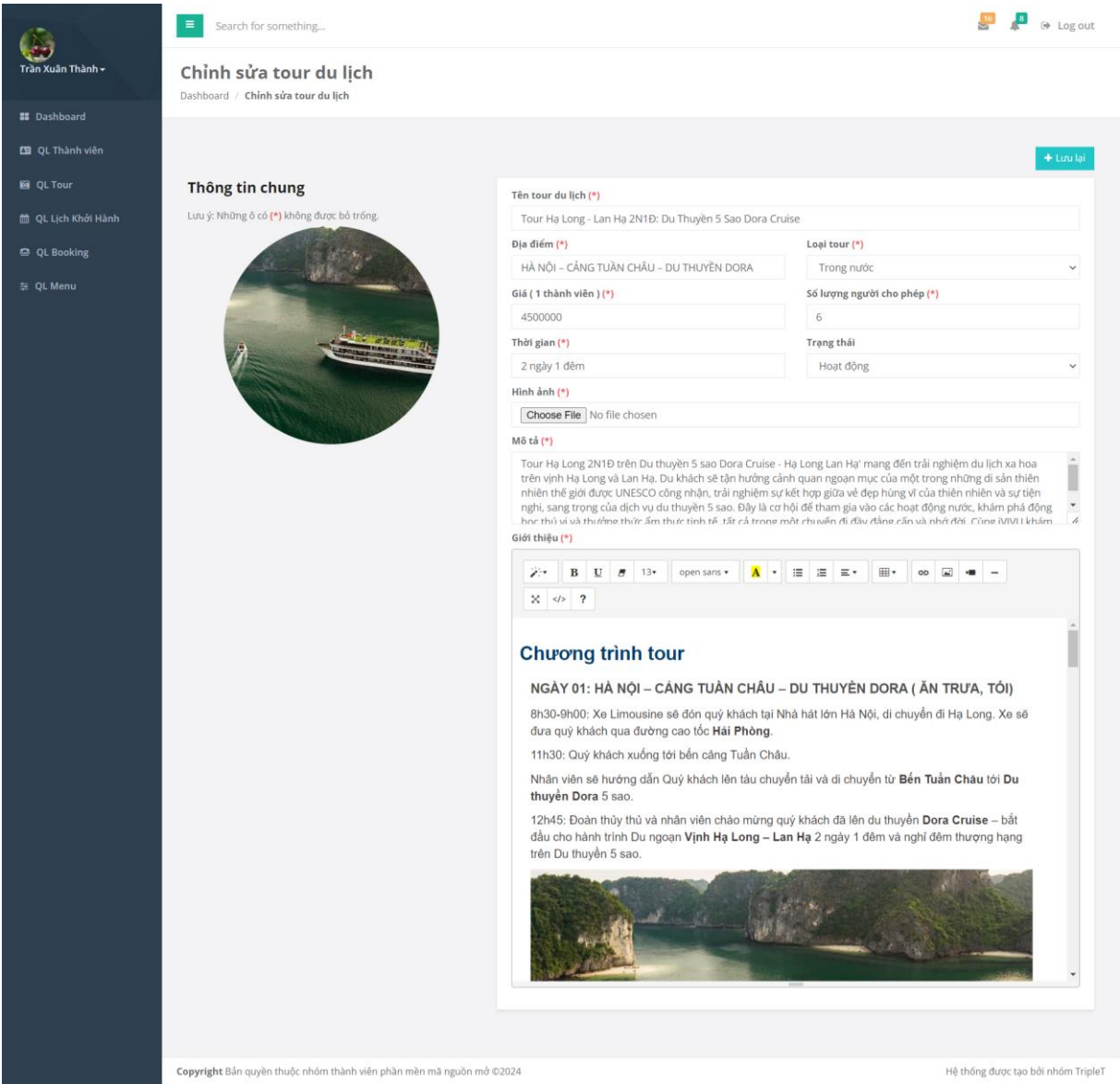


Hình 15. Giao diện danh sách tour du lịch

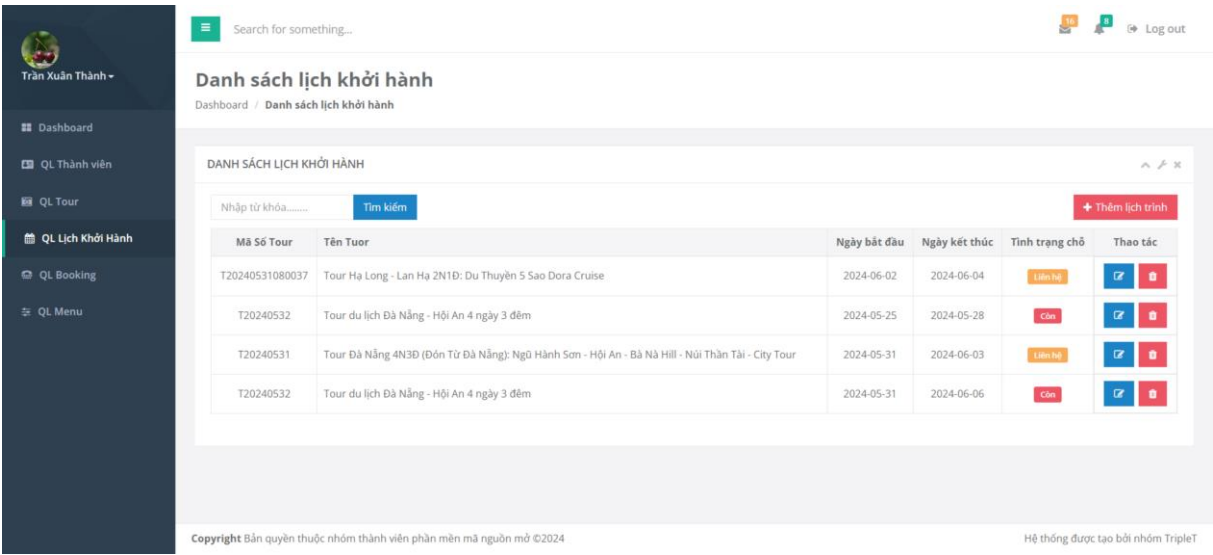


Hình 16. Giao diện thêm tour du lịch





Hình 17. Giao diện chỉnh sửa tour du lịch



Hình 18. Giao diện danh sách lịch khởi hành

Trần Xuân Thành ▾

- Dashboard
- QL Thành viên
- QL Tour
- QL Lịch Khởi Hành
- QL Booking
- QL Menu

Search for something...

Log out

Thêm lịch khởi hành

Dashboard / Thêm lịch khởi hành

Thông tin chung

Nhập thông tin của người sử dụng  
 Lưu ý: Những trường đánh dấu (\*) là bắt buộc

Tên tour du lịch (\*)

Chọn tour ▾

Ngày bắt đầu (\*)

mm/dd/yyyy

Ngày về (\*)

mm/dd/yyyy

Tình trạng chỗ (\*)

Còn ▾

Lưu lại

Copyright Bản quyền thuộc nhóm thành viên phần mềm mã nguồn mở ©2024

Hệ thống được tạo bởi nhóm TripleT

Hình 19. Giao diện thêm lịch khởi hành

Trần Xuân Thành ▾

- Dashboard
- QL Thành viên
- QL Tour
- QL Lịch Khởi Hành
- QL Booking
- QL Menu

Search for something...

Log out

Chỉnh sửa lịch trình

Dashboard / Chỉnh sửa lịch trình

Thông tin chung

Nhập thông tin của người sử dụng  
 Lưu ý: Những trường đánh dấu (\*) là bắt buộc

Tên tour du lịch (\*)

Tour Hạ Long - Lan Hạ 2N1Đ: Du Thuyền 5 Sao Dora Cruise ▾

Ngày bắt đầu (\*)

06/02/2024

Ngày về (\*)

06/04/2024

Tình trạng chỗ (\*)

Liên hệ ▾

Lưu lại

Copyright Bản quyền thuộc nhóm thành viên phần mềm mã nguồn mở ©2024

Hệ thống được tạo bởi nhóm TripleT

Hình 21. Giao diện Sửa lịch khởi hành

Trần Xuân Thành ▾

- Dashboard
- QL Thành viên
- QL Tour
- QL Lịch Khởi Hành
- QL Booking
- QL Menu

Search for something...

Log out

Danh sách đặt tour

Dashboard / Danh sách đặt tour

DANH SÁCH ĐẶT TOUR

Chọn tình trạng ▾

Nhập từ khóa.....

Tìm kiếm

Mã tour ☞	Tên khách hàng ☞	Điện thoại ☞	số lượng người ☞	Tình trạng ☞	Thao tác ☞
> T20240531080037	thiê	0124587452	6	Chưa duyệt	<div></div> <div></div>
> T20240532	Trần Xuân Thành	0936867190	8	Liên hệ lại	<div></div> <div></div>
> T20240532	Trần Xuân Thành	0936867190	5	Chưa duyệt	<div></div> <div></div>
> T20240532	Trần Xuân Thành	0936867190	2	Chưa duyệt	<div></div> <div></div>

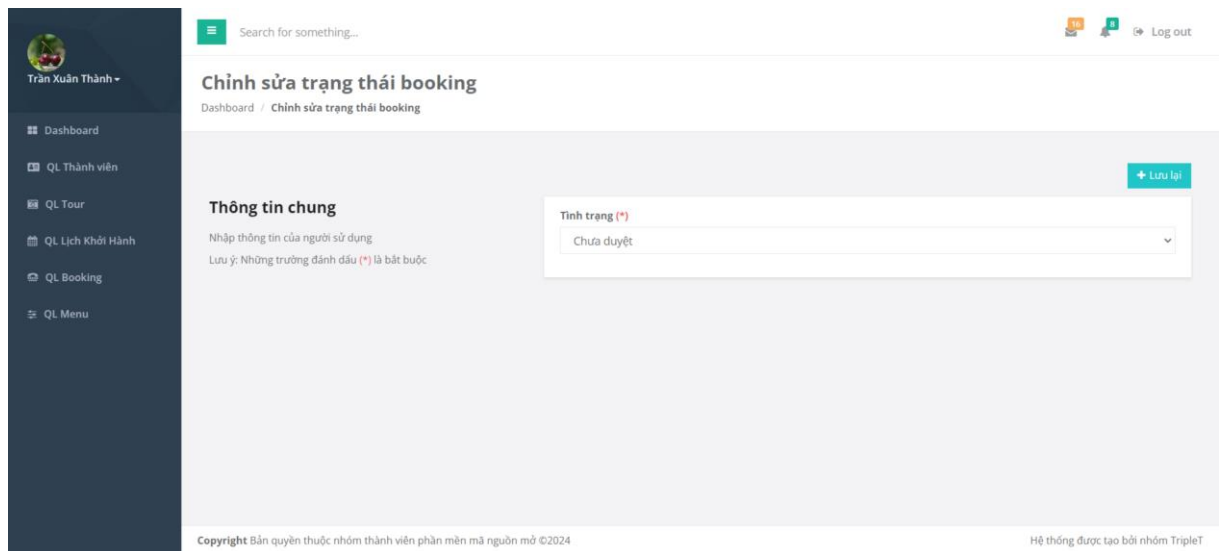
Copyright Bản quyền thuộc nhóm thành viên phần mềm mã nguồn mở ©2024

Hệ thống được tạo bởi nhóm TripleT

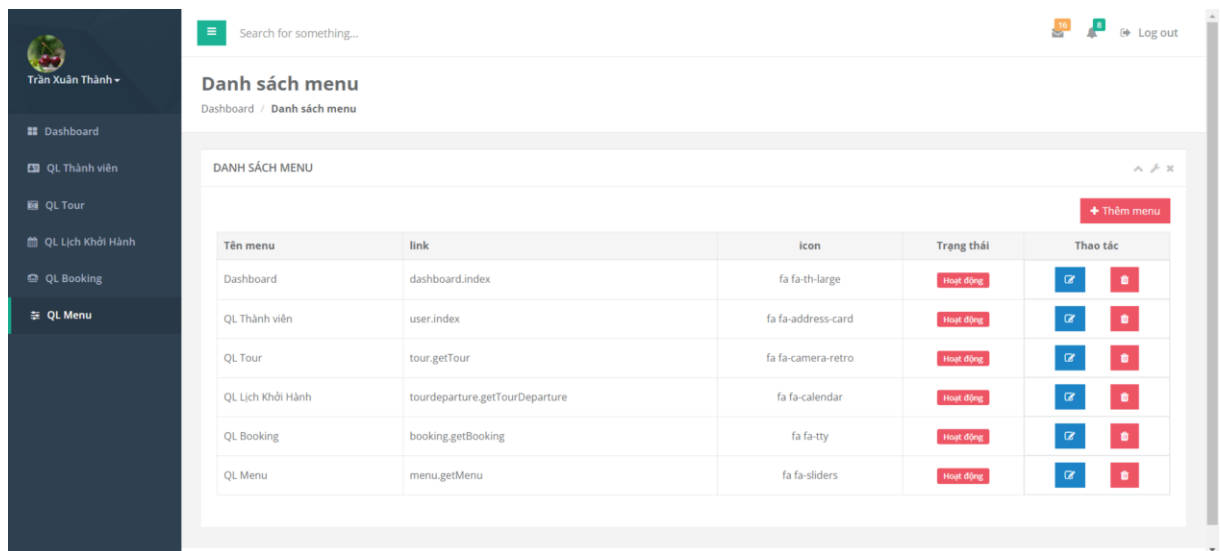
Hình 22. Giao diện danh sách đặt tour

55

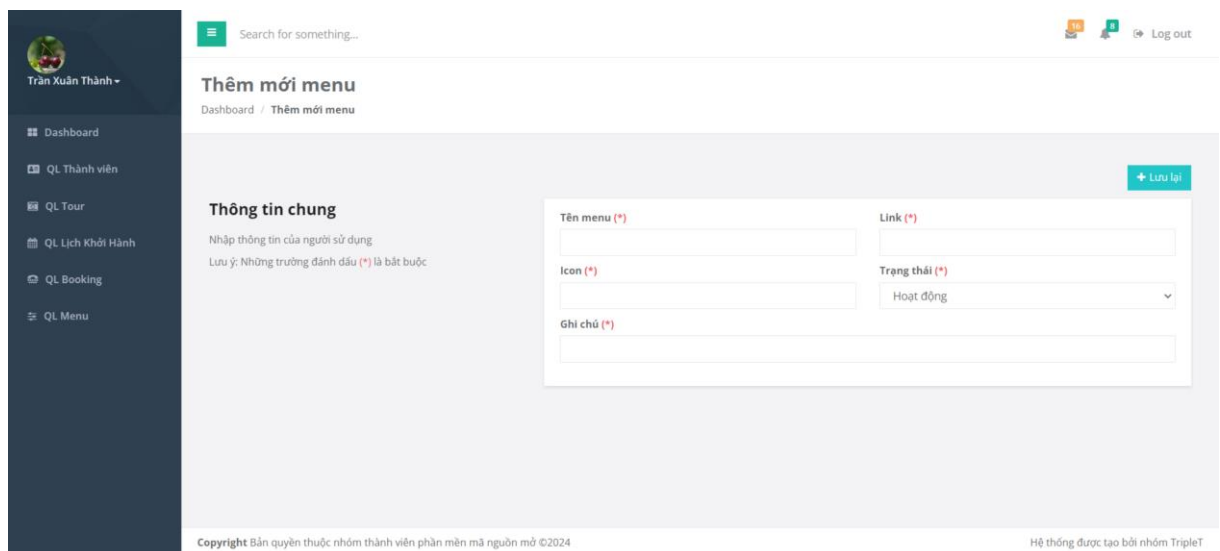




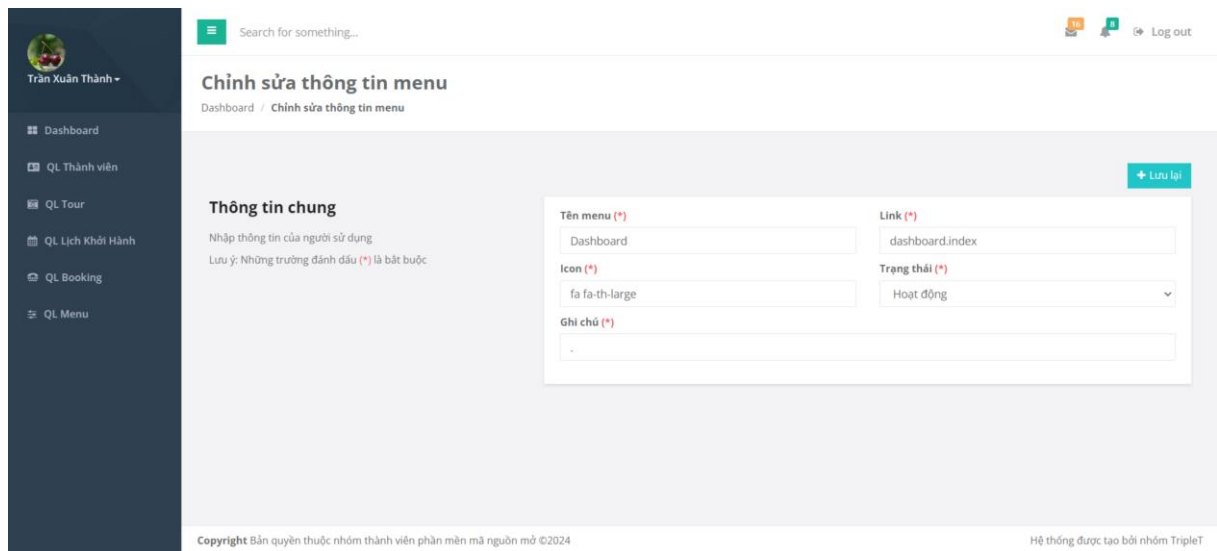
Hình 23. Giao diện Sửa trạng thái booking



Hình 24. Giao diện danh sách item menu

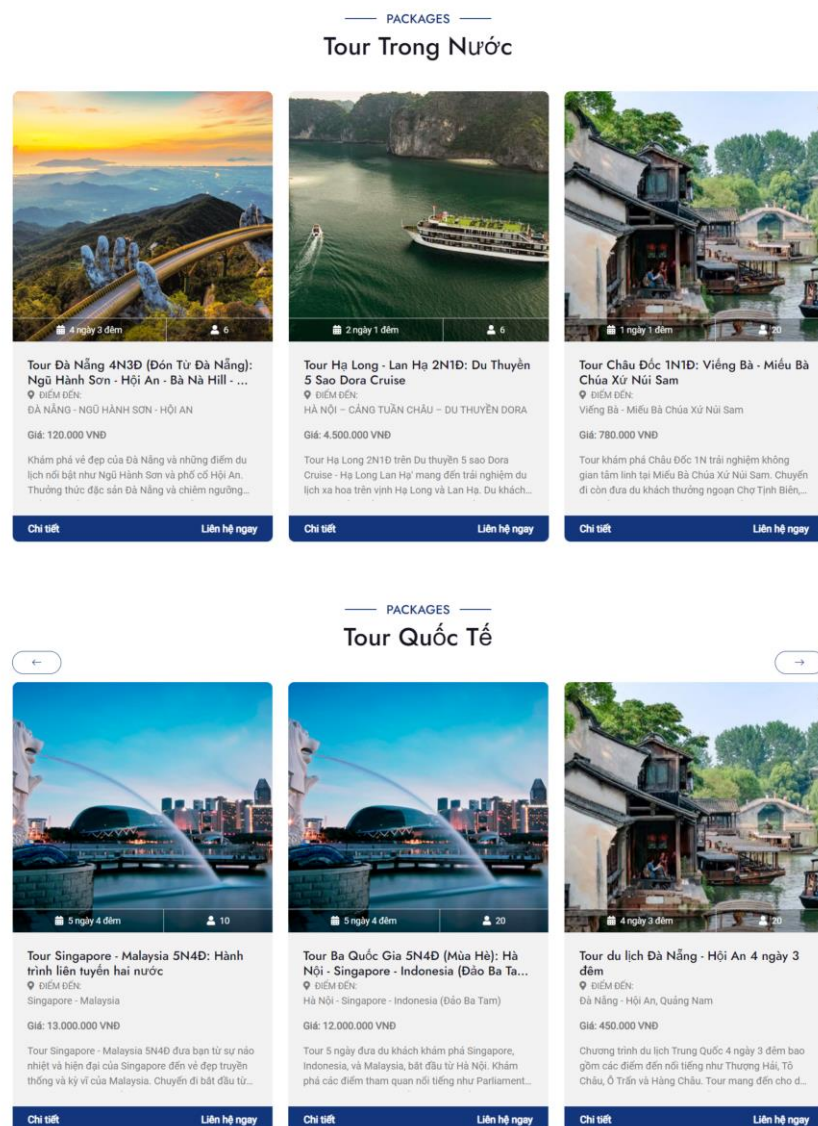


Hình 25. Giao diện thêm item menu



Hình 26. Giao diện sửa item menu

### 3.3. Giao diện các trang cho người dùng



Hình 27. Giao diện tour hiện hành

## Tour Châu Đốc 1N1Đ: Viếng Bà - Miếu Bà Chúa Xứ Núi Sam

Mã tour: T20240602155851



### Tour Châu Đốc 1N1Đ: Viếng Bà - Miếu Bà Chúa Xứ Núi Sam

Tour khám phá Châu Đốc 1N trải nghiệm không gian tâm linh tại Miếu Bà Chúa Xứ Núi Sam. Chuyển đi còn đưa du khách thưởng ngoạn Chợ Tịnh Biên, Núi Cấm và Rừng Tràm Trà Sư, nơi cảnh quan thiên nhiên và văn hóa địa phương hòa quyện. Hành trình này hứa hẹn nhiều trải nghiệm đáng nhớ với ẩm thực đặc sắc của vùng sông nước. Cùng iVIVU khám phá ngay hôm nay!

### Chương trình tour

#### TỐI NGÀY 01: TP. HỒ CHÍ MINH - TP. CHÂU ĐỐC (NGHỈ ĐÊM TRÊN XE)

22h00 - 23h00: Xe và hướng dẫn viên Công ty du lịch đón quý khách tại điểm hẹn, khởi hành đi Châu Đốc. Nghỉ đêm trên xe.

#### NGÀY 01: MIẾU BÀ CHÚA XỨ - NÚI CẤM/RỪNG TRÀM TRÀ SƯ - TP.HCM (ĂN SÁNG, TRƯA)

04h00: Đến Châu Đốc, quý khách dừng chân nghỉ ngơi, vệ sinh cá nhân sau đó viếng **Miếu Bà Chúa Xứ Núi Sam** nổi tiếng hiển linh, viếng **Tây An Cổ Tự, Lăng Thoại Ngọc Hầu** - Người có công khai mở đất An Giang.



Miếu Bà Chúa Xứ: điểm tâm linh linh thiêng, Vẻ đẹp kiến trúc huyền bí.

06h30: Quý khách dùng điểm tâm sáng buffet (nếu trường hợp quá tải trong mùa cao điểm thì sẽ ăn theo dạng tô ly).

Sau đó đoàn tham quan và mua sắm tại **Chợ Tịnh Biên**.



### Lịch Trình và Giá Tour

Đặt ngay, chỉ 2 phút. Hoặc gọi (028) 3933 8002

Chọn Lịch Trình và Xem Giá:

Số lượng người: x 780,000 - 1 +

Liên hệ để xác nhận chỗ

Tổng Giá Tour: 780,000 VNĐ

Tên của bạn:

Số điện thoại:

Email:

Ghi chú:

Liên hệ ngay

Đặt chỗ ngay

Hình 28. Giao diện chi tiết tour và booking tour

The screenshot shows the login interface of 'The Sun Travel' website. The header includes social media icons (Twitter, Facebook, LinkedIn, Instagram, YouTube) and links for 'Register' and 'Login'. The main navigation bar contains 'Trang chủ', 'Tour', 'Giới thiệu', 'Dịch vụ', 'Blog', and 'Liên hệ'. The login form is centered and titled 'Đăng nhập'. It contains fields for 'Email (\*)' and 'Password (\*)', both with placeholder text 'Nhập email của bạn' and 'Nhập mật khẩu của bạn' respectively. There is a 'Remember Me' checkbox. Below the fields is a dark blue 'Đăng nhập' button. At the bottom of the form, there are links for 'Quên mật khẩu?' and 'Bạn chưa có tài khoản? Đăng ký'.

Hình 29. Giao diện đăng nhập

The screenshot shows the registration interface of 'The Sun Travel' website. The header and navigation bar are identical to the login page. The registration form is titled 'Đăng ký tài khoản'. It includes fields for 'Họ và tên (\*)' (placeholder: 'Nhập họ tên của bạn'), 'Email (\*)' (placeholder: 'Nhập email của bạn'), 'Mật khẩu (\*)' (placeholder: 'Nhập mật khẩu của bạn'), and 'Nhập lại mật khẩu (\*)' (placeholder: 'Nhập lại mật khẩu của bạn'). There is a checkbox for 'Tôi đồng ý với các điều khoản'. Below the fields is a dark blue 'Đăng ký' button. At the bottom of the form, there is a link for 'Bạn đã có tài khoản? Đăng nhập'.

Hình 30. Giao diện đăng ký

## KẾT LUẬN

Trong quá trình phát triển hệ thống phần mềm quản lý tour du lịch bằng Laravel, chúng em đã đạt được nhiều kết quả tích cực. Hệ thống đã hoàn thiện các chức năng chính như quản lý thông tin khách hàng, tour du lịch, đặt tour. Những chức năng này giúp tối ưu hóa quy trình quản lý và nâng cao trải nghiệm khách hàng.

- **Những đóng góp chính:**

Tính năng toàn diện: Hệ thống cung cấp giải pháp toàn diện cho việc quản lý tour du lịch.

Công nghệ tiên tiến: Sử dụng Laravel giúp đảm bảo bảo mật, dễ bảo trì và mở rộng.

Trải nghiệm người dùng: Giao diện thân thiện, dễ sử dụng.

Khả năng mở rộng: Dễ dàng tùy chỉnh và mở rộng theo yêu cầu.

- **Khó khăn và hạn chế:**

Chúng em gặp phải một số khó khăn như độ phức tạp của hệ thống và nguồn lực phát triển hạn chế.

- **Định hướng phát triển:**

*Tích hợp AI và Machine Learning.*

*Phát triển ứng dụng di động.*

*Nâng cấp bảo mật.*

*Tăng cường tương tác khách hàng.*

Hệ thống quản lý tour du lịch bằng Laravel đã mang lại nhiều lợi ích thiết thực, góp phần nâng cao hiệu quả quản lý và chất lượng dịch vụ. Với những cải tiến liên tục, hệ thống sẽ ngày càng hoàn thiện và đáp ứng tốt hơn nhu cầu của khách hàng và công ty.

## TÀI LIỆU THAM KHẢO

- [1] Slide\_OSSD, phát triển phần mềm mã nguồn mở, TS. Nguyễn Văn Thắm
- [2] <https://laravel.com/docs/10.x/installation>
- [3] <https://getbootstrap.com/docs/5.3/getting-started/introduction/>
- [4] <https://themewagon.com/>
- [5] [http://webappplayers.com/inspinia\\_admin-v2.9.4/](http://webappplayers.com/inspinia_admin-v2.9.4/)