TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

ĐẠI HỌC QUỐC GIA HÀ NỘI



*Báo cáo môn tích hợp hệ thống*

*Lớp k57 T - Nhóm 5*

*Năm học : 2015 – 2016*

Giảng viên : TS. Nguyễn Ngọc Hóa

Sinh viên : Phan Thị Tâm

Đào Huy Hoàng

Đỗ Thanh Tuyền

Nguyễn Ngọc Bảo

MỤC LỤC

[I. Giới thiệu 1](#_Toc439453377)

[1. GCO Smile Odonto 1](#_Toc439453378)

[2. Chikitsa 3](#_Toc439453379)

[3. Hospice Volunteer Solutions 3](#_Toc439453380)

[II. Công nghệ sử dụng 6](#_Toc439453381)

[III. Chức năng hệ thống tích hợp 3 module 6](#_Toc439453382)

[1. Login: 6](#_Toc439453383)

[2. Logout module: 9](#_Toc439453384)

[IV. Chi tiết code và hướng giải quyết bài toán. 16](#_Toc439453385)

[1. Module tổng: 16](#_Toc439453386)

[2. Làm sao để cho khi đăng nhập trang chính thì 3 module khác tự động đăng nhập. 19](#_Toc439453387)

[3. Một module logout thì tất cả các module khác đều logout và login một module thì các module còn lại đều login. 22](#_Toc439453388)

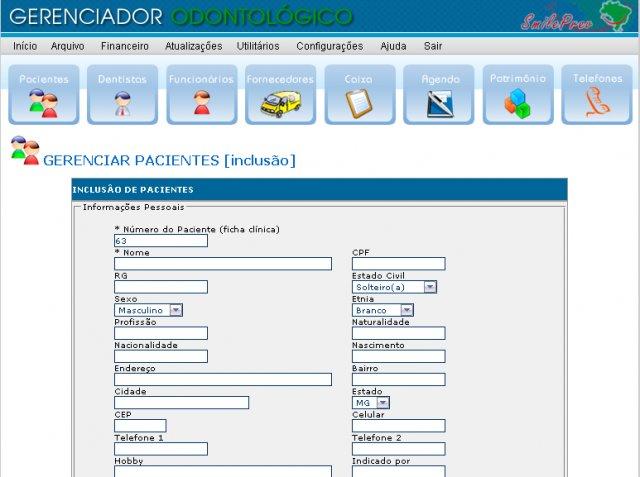
1. Giới thiệu

Với mục đích tích hợp hệ thống 3 module trong y học, nhóm em giới thiệu chức năng 3 module mã nguồn mở: gco-smile, chikitsa và hospice-volunteer-solutions.

1. GCO Smile Odonto

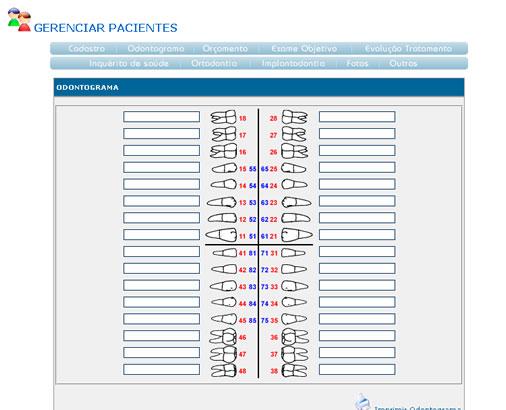
Module GCO quản lý phòng khám nha khoa

* Quản lý hồ sơ bệnh nhân và nha sỹ



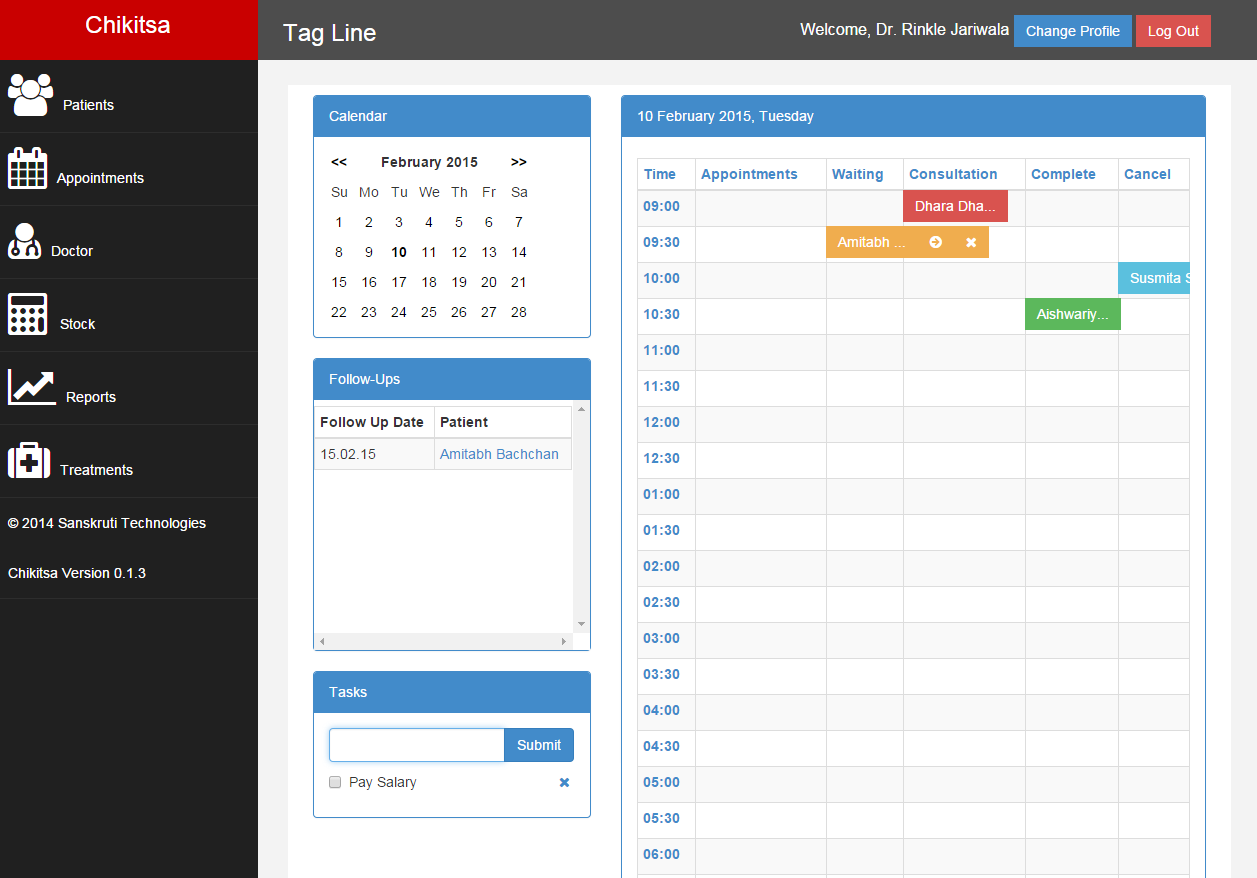


* Chụp x – quang răng



1. Chikitsa

* Module Chikitsa là một hệ thống quản lý bệnh nhân giúp theo dõi, giữ liên lạc với các bệnh nhân, đặt lịch hẹn, in phiếu thanh toán và quản lý thuốc.
* Yêu cầu : PHP 5.5



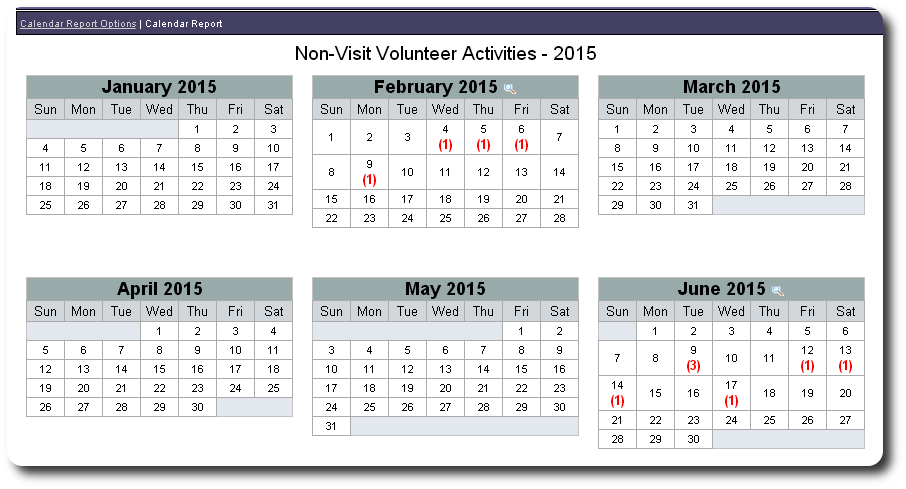
1. Hospice Volunteer Solutions

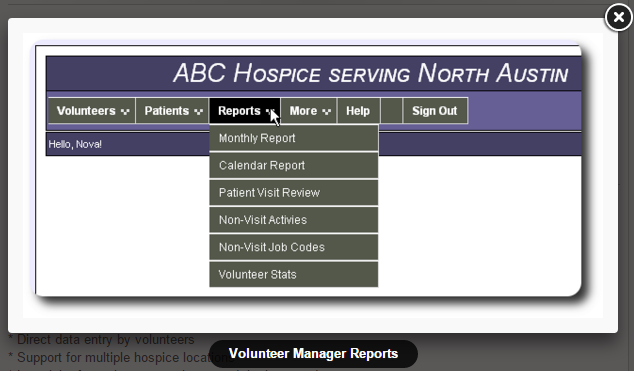
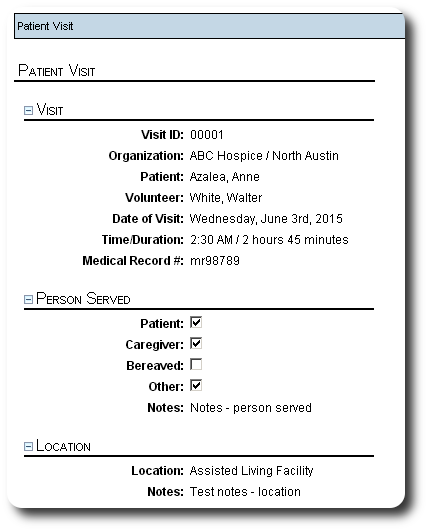
Module Hospice Volunteer Solutions là một hệ thống quản lý các tình nguyên viên quản lý viện chăm sóc đặc biệt.

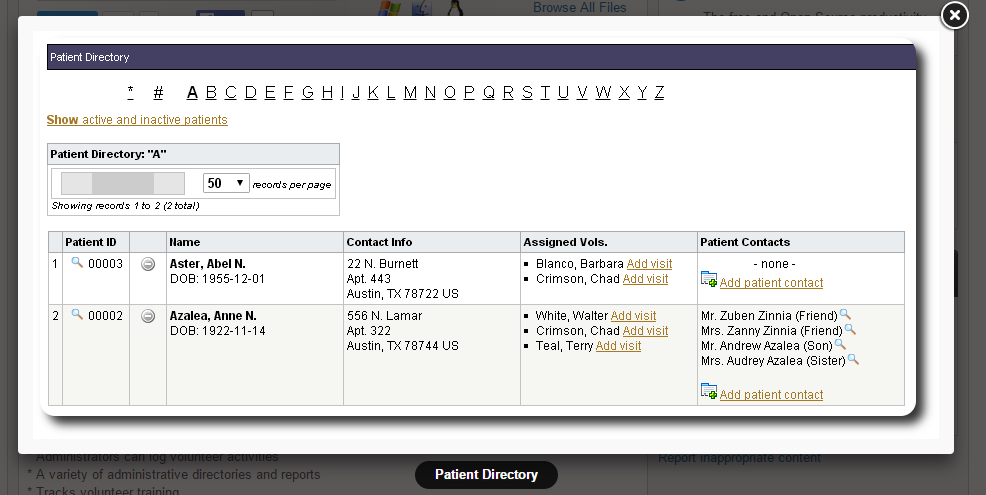
Chương trình cung cấp các tính năng sau:

* Nhập dữ liệu trực tiếp từ tình nguyện viên
* Hỗ trợ nhiều địa điểm chăm sóc đặc biệt
* Chức năng ghi lại các lần thăm cho bệnh nhân, người chăm sóc và gia quyến
* Ghi lại các hoạt động của tình nguyện viên như: hỗ trợ quản lý, lên kế hoạch sự kiện và giúp đỡ văn phòng
* Người quản lý có thể ghi lại các hoạt động của tình nguyện viên
* Một loạt các thư mục và báo cáo hành chính
* Các theo dõi đào tạo tình nguyện viên









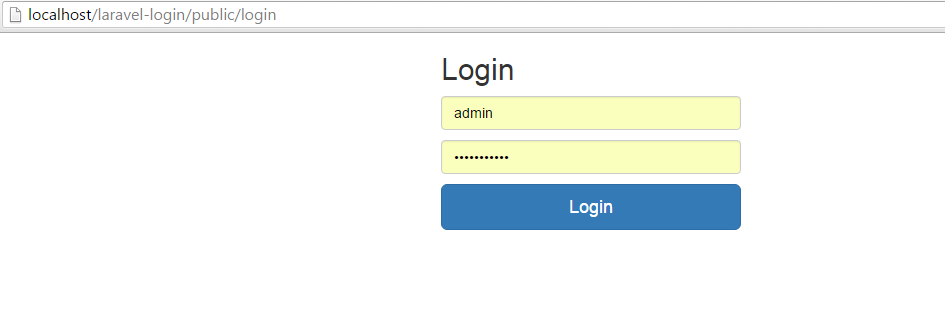
1. Công nghệ sử dụng

* MySQL
* Php – Laravel
* Sản phẩm được đẩy lên Github:

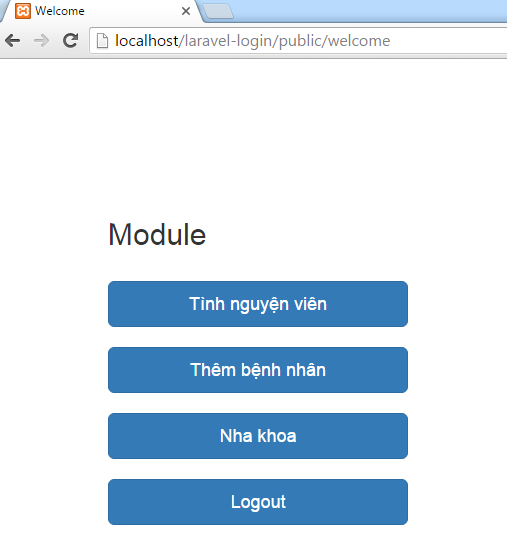
*https://github.com/nhom5UET/tichhophethong*

1. Chức năng hệ thống tích hợp 3 module
2. Login:

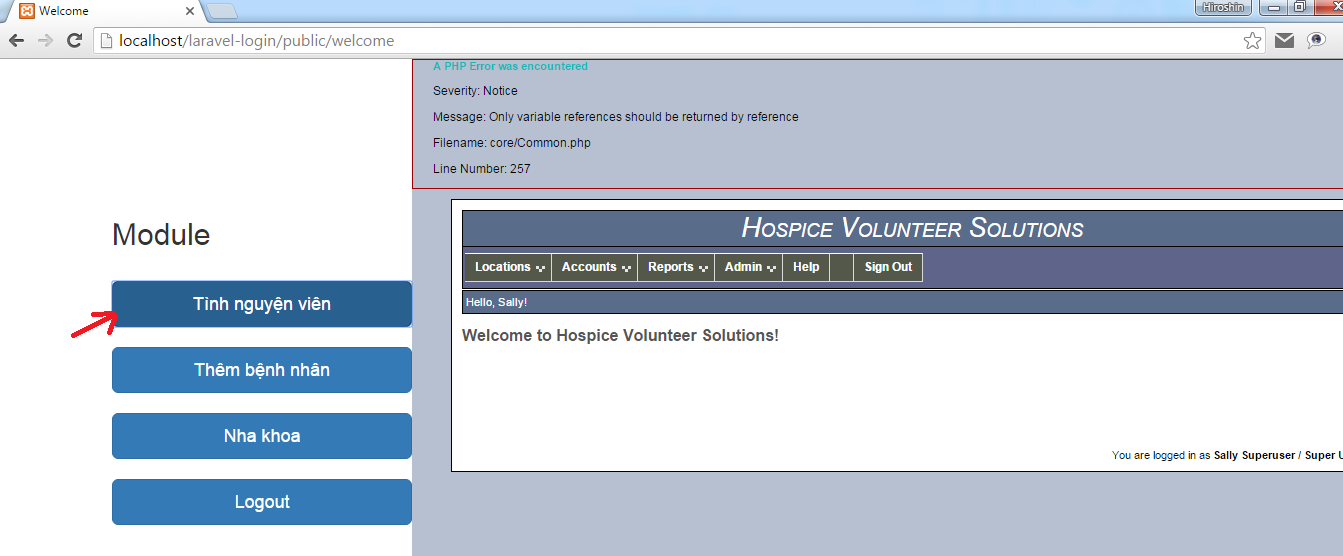
Hình 1



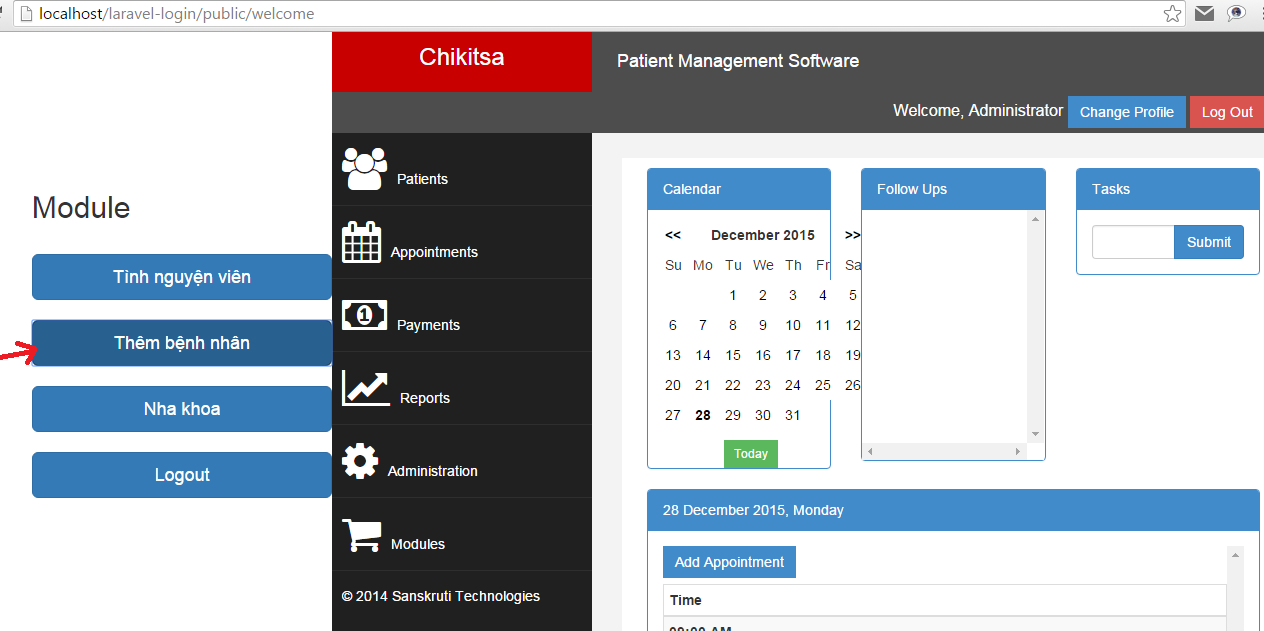
Hình 2



Hình 2.1



Hình 2.2



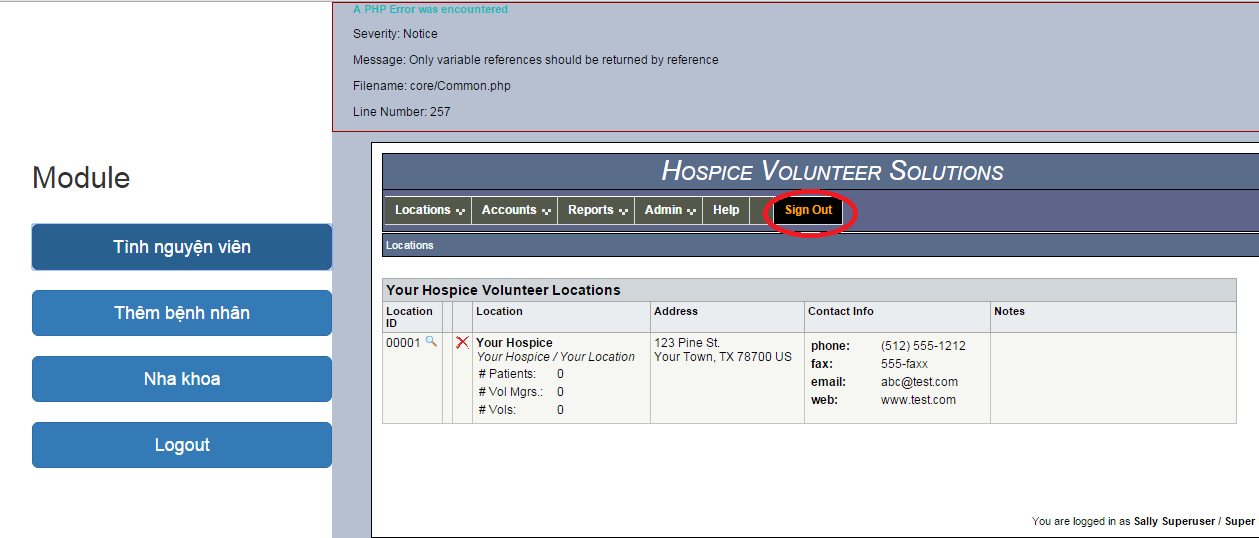
Hình 2.3



1. Logout module:

- Logout Tình nguyện viên thì các module còn lại đều logout

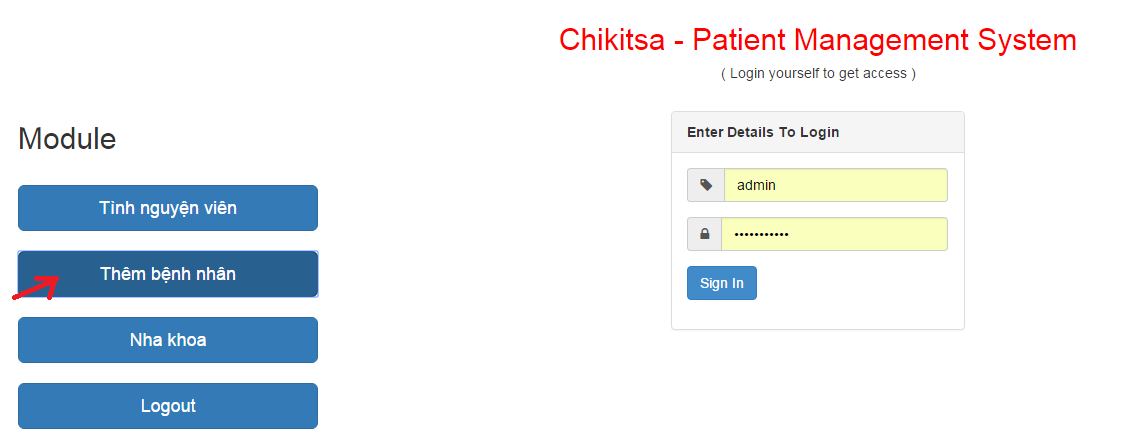
Hình 3.1



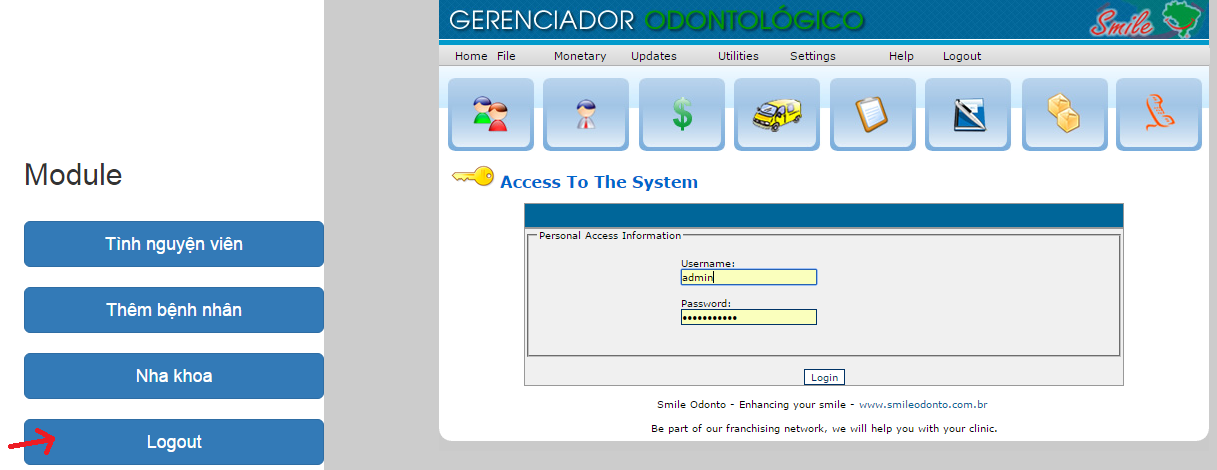
Hình 3.2:



Hình 3.3:

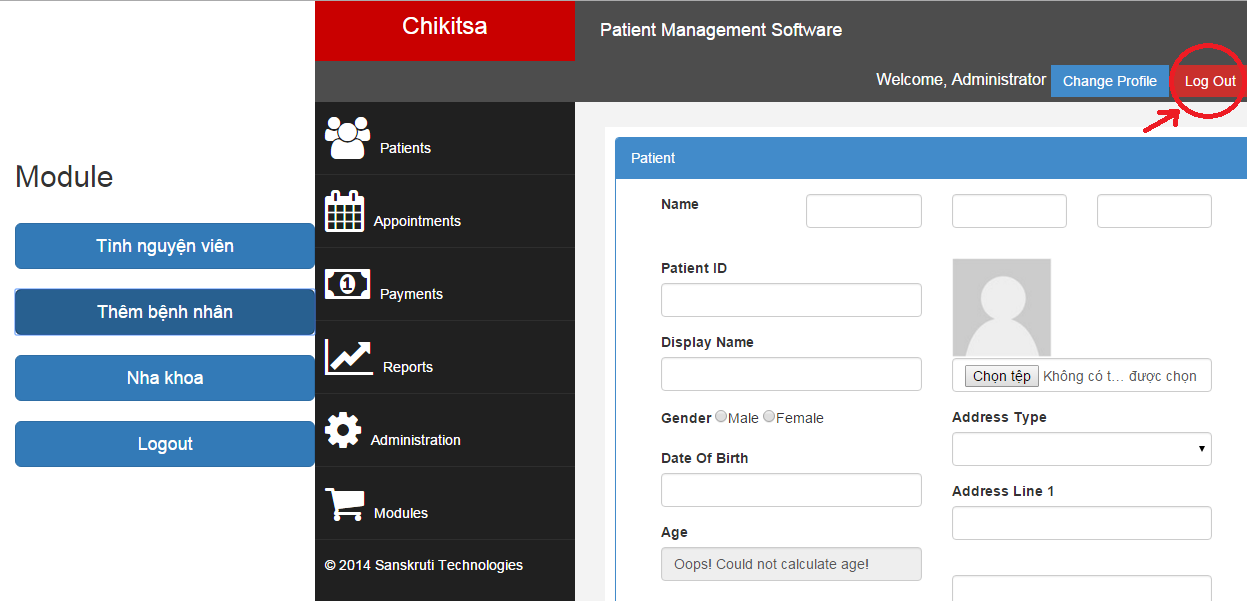


Hình 3.4:

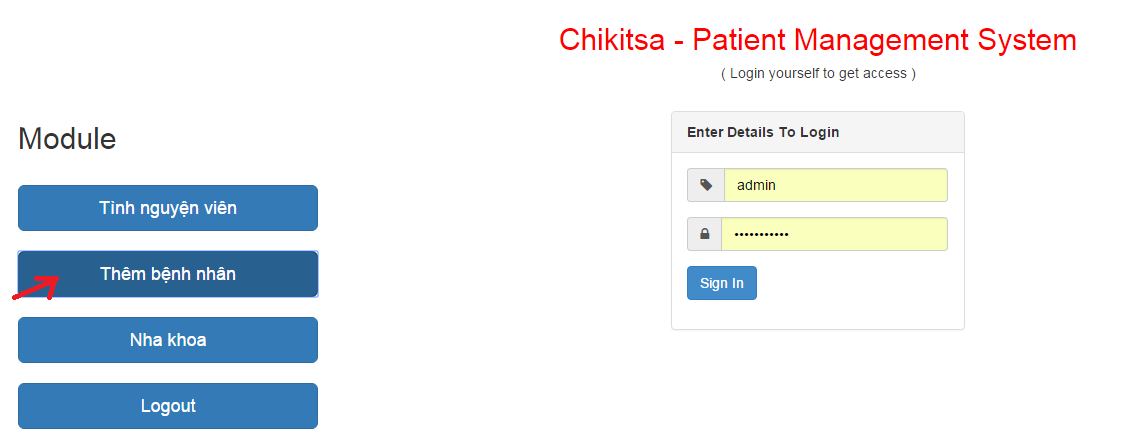


* Logout Thêm bệnh nhân thì các module khác đều log out

Hình 4.1:



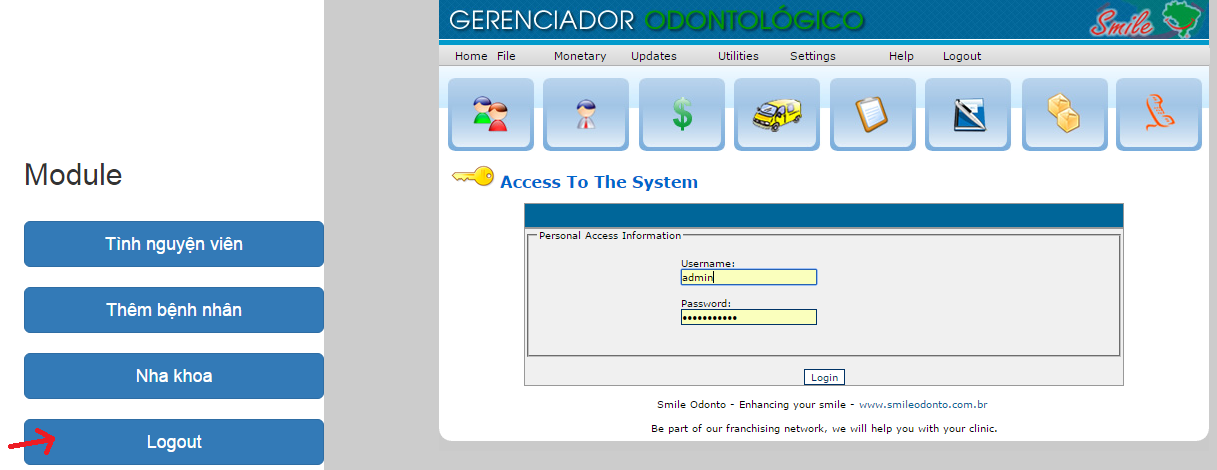
Hình 4.2



Hình 4.3:



Hình 4.4:

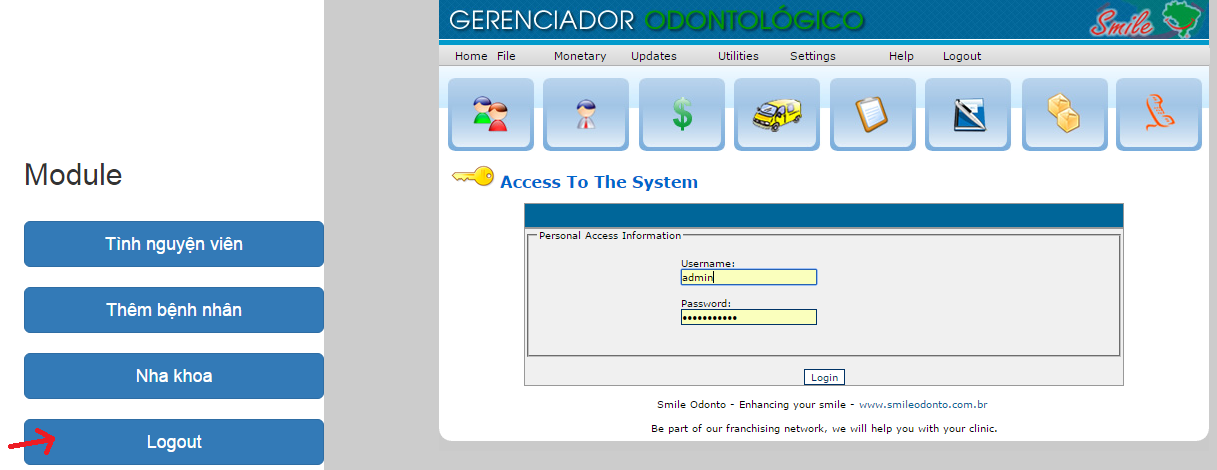


* Logout Nha khoa thì các module còn lại đều logout

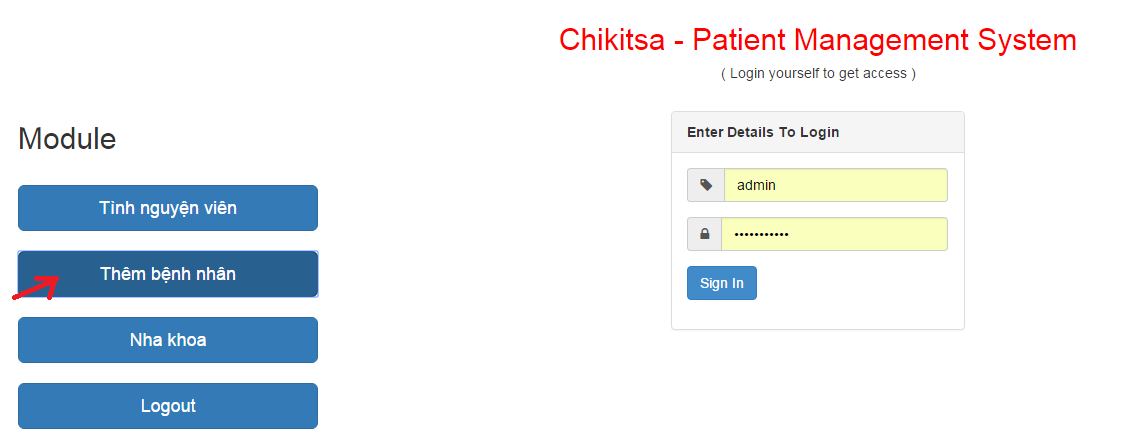
Hình 5.1:



Hình 5.2:



Hình 5.3:

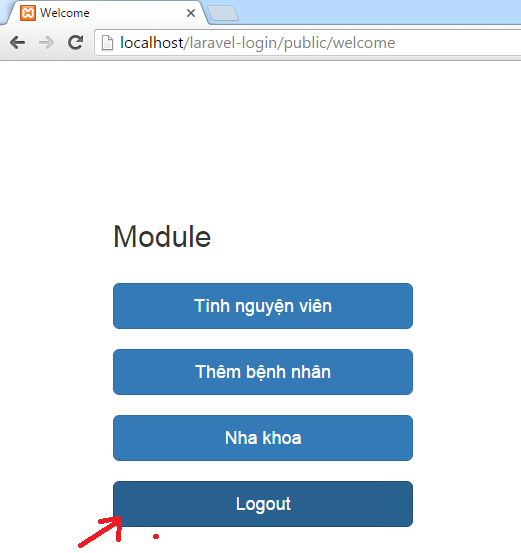


Hình 5.4:

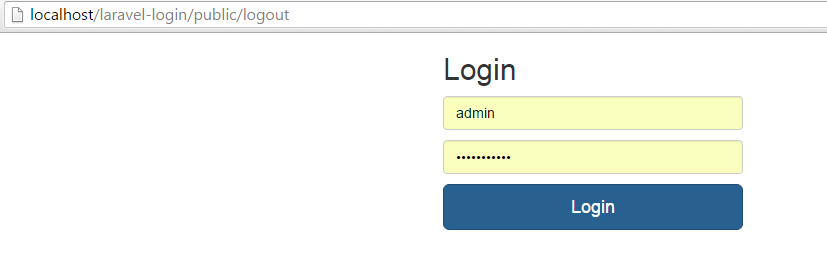


* Login một module thì tất cả module còn lại đều login.
* Logout thì sẽ quay về login

Hình 5.5



Hình 5.6



1. Chi tiết code và hướng giải quyết bài toán.
2. Module tổng:

Tạo 2 file module.blade.php trong ../laravel-login/app/views/ với các nội dung sau:

Hàm khởi tạo cho iframe:



Form của các module:



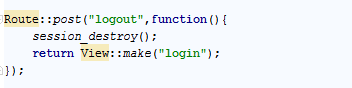
Tạo Route đăng ký:

Tạo file register.blade.php có nội dung như trong thư mục ../laravel-login/app/views/register.blade.php.

Sau đó tạo trong route như hình sau để kiểm tra đăng ký, đăng nhập và tạo session khi vừa đăng nhập:







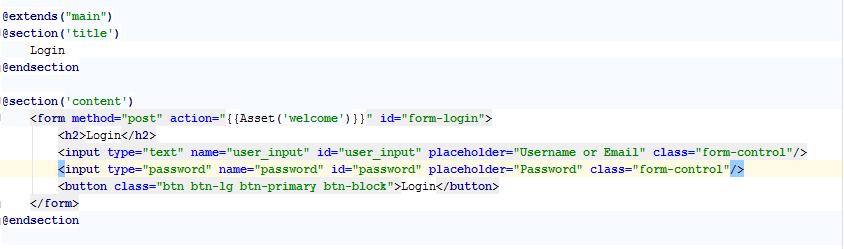
Hàm main.blade.php:



Check login and create require:



File Login.blade.php:



1. Làm sao để cho khi đăng nhập trang chính thì 3 module khác tự động đăng nhập.

Khi đăng nhập thành công trang chính sẽ tạo ra 2 sesion: $\_SESSION[“session\_user”] và $\_SESSION[“session\_password”].



Tạo file check\_login.php trong thư mục ../laravel-login/app/models/ với nội dung như sau:



3 modules sẽ kiểm tra xem nếu người dùng đã đăng nhập thì sẽ cho phép đăng nhập vào hệ thống với user, password được gán cố định cho từng modules.

Module HVOL:

Để sử dụng được function check\_login() của laravel-login ta cần gán lệnh sau vào file hvol/application/controllers/login.php:



Include("../laravel-login/app/models/check\_session.php").

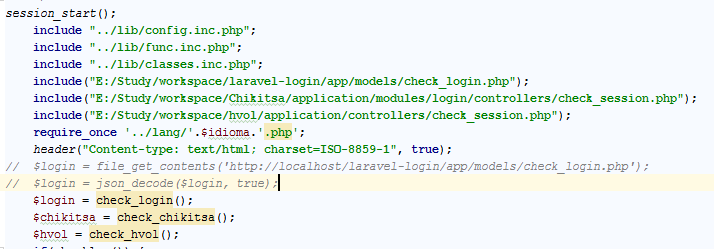
Nếu tồn tại login['check\_session'] thì gán cố định user, password để cho module đăng nhập.



Tương tự với 2 modules: Chikitsa và Gco\_smile:

Gco\_smile:

Gán vào file *../gco\_smile/wallpapers/index\_ajax.php*





Chikitsa:

Gán vào file *../Chikitsa/application/modules/login/controllers/Login.php*

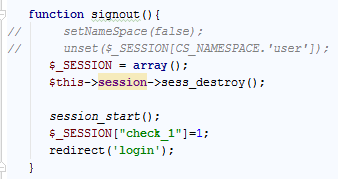


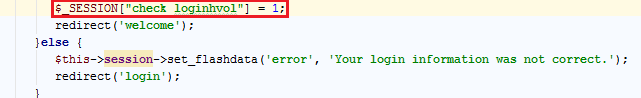


1. Một module logout thì tất cả các module khác đều logout và login một module thì các module còn lại đều login.

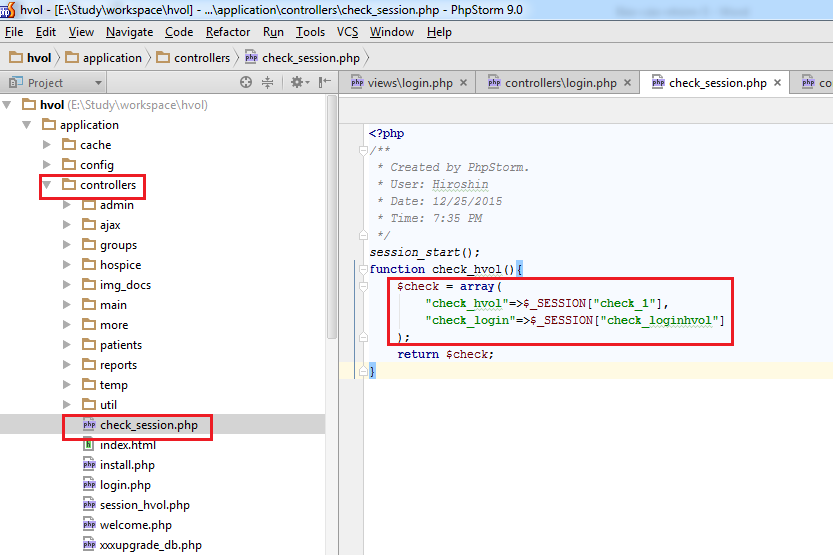
Tạo một Session ngay sau khi logout, login một module nào đó.

Ví dụ tạo một $\_SESSION["check\_1"]=1 và $\_SESSION["check\_loginhvol"] = 1 sau khi logout hoặc login HVOL.





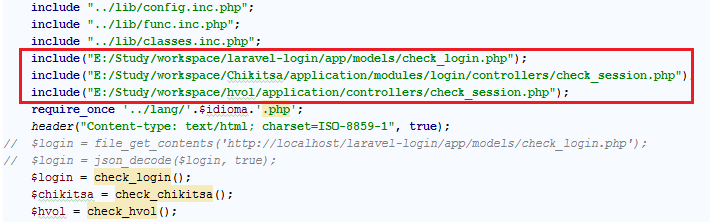
Lưu các biến này vào một mảng trong file check\_session.php



Sau đó, Chikitsa và Gco\_smile sẽ kiểm tra nếu tồn tại Session sau khi login thì sẽ gán cứng user hoặc password để cho các module này login, hoặc nếu logout thì sẽ trở về trang đăng nhập.

Ví dụ trong Gco\_smile:

Include file check\_session.php của hvol:



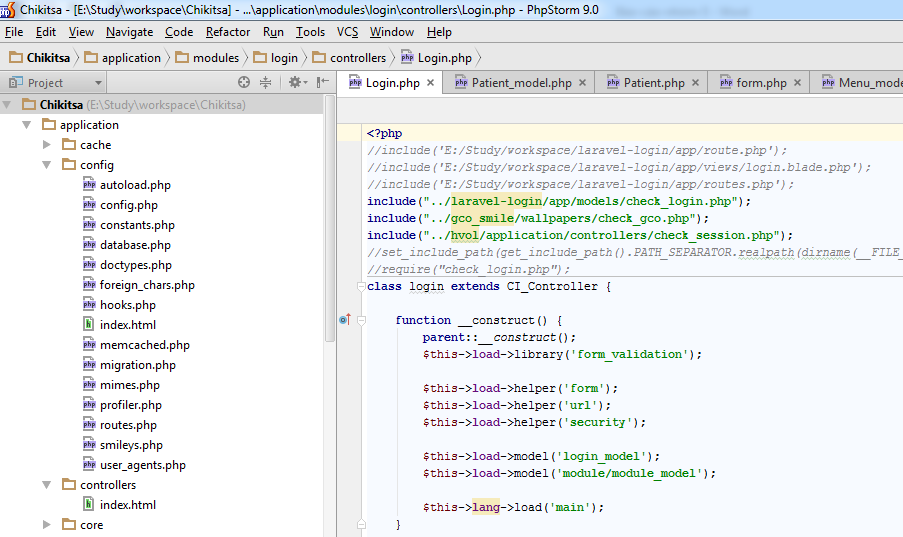
Kiểm tra logout:



//form login.

Ví dụ với Chikitsa, sau khi module hvol đăng nhập thì nó sẽ kiểm tra nếu tồn tại $\_hvol[“check\_login”] thì sẽ gán cứng user và password để đăng nhập.

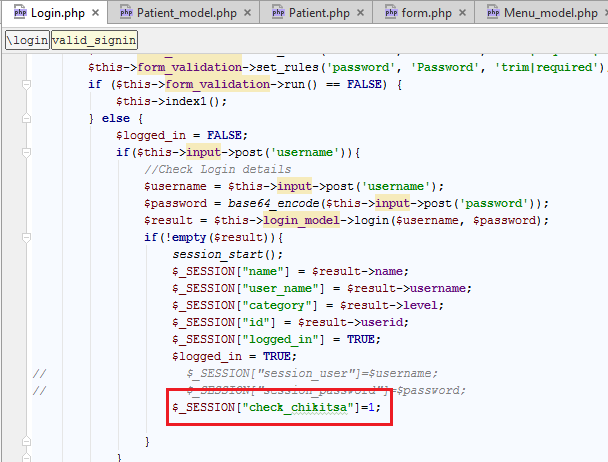
Include các file của 2 module còn lại:

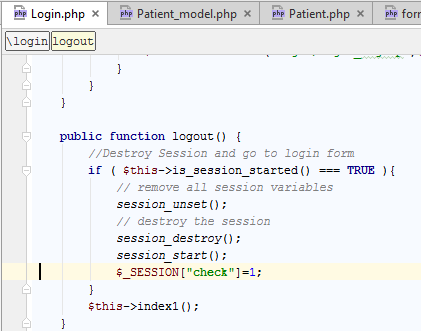




Tương tự gán cho các module còn lại.

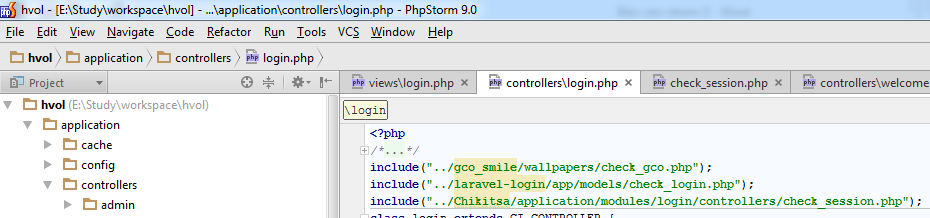
CHIKITSA:







HVOL:



Chú ý: nếu muốn đăng tạo một người dùng mới để đăng nhập và được các chức năng như trên thì bỏ phần sau trong Gco\_smile để thỏa mãn yêu cầu:

