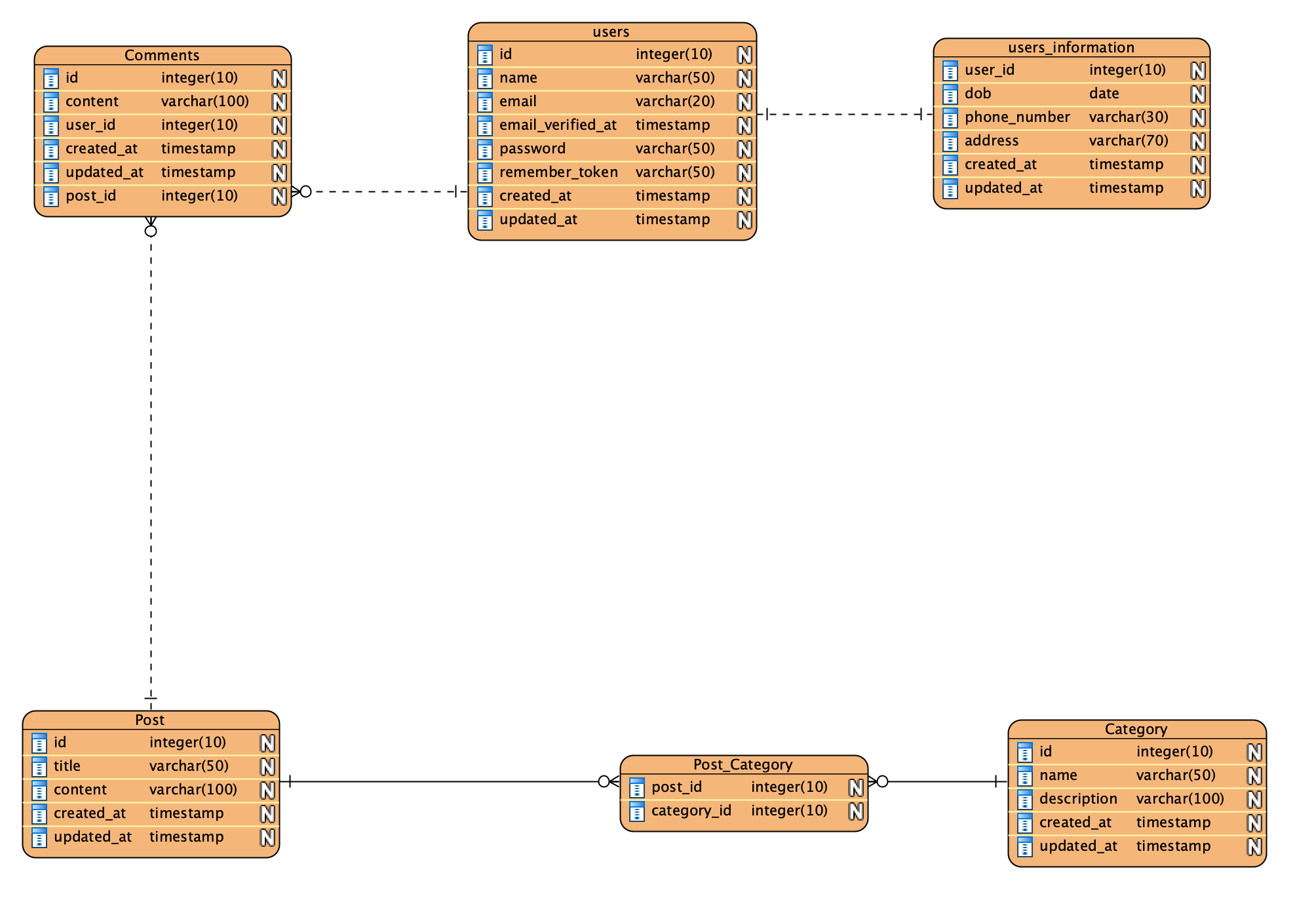
# Exercises

## Bài tập 1: Sử dụng Laravel migrations để thực hiện bài toán sau

Viết migrations cho các bảng sau dựa theo sơ đồ ER diagram



Đối với các bảng đã tạo migration sẵn thì không cần phải tạo migrations nữa, chỉ cần tạo các bảng chưa có.

## Bài tập 2: Sử dụng Eloquent relationship để thực hiện xây mối quan hệ (1-1, 1-M, M-N) dựa theo hình vẽ ở trên

Thực hiện relationship 1-1, 1-M như sau

Post----Many – Many ----Category ( Một **Post** thuộc nhiều **Category**, 1 **Category** nhiều **Post**)

User --- 1 – Many ----Comments (Một **User** nhiều **Comments**, một **Comment** thuộc 1 **User**)

Post --- 1 – Many --- Comments (Một **Post** nhiều **Comments**, một **Comment** trong 1 **Post** duy nhất)

## Bài tập 3: Sử dụng authentication trong Laravel, thực hiện chức năng Đăng nhập và đăng kí người dùng như sau

Xây dựng giao diện cho phép người dùng đăng nhập và đăng kí thành viên trong hệ thống. Giao diện cần phải khác với giao diện ở trong bài để thực hành làm việc với view trong Laravel.

Thực hiện cả chức năng Remember checked có hay không.

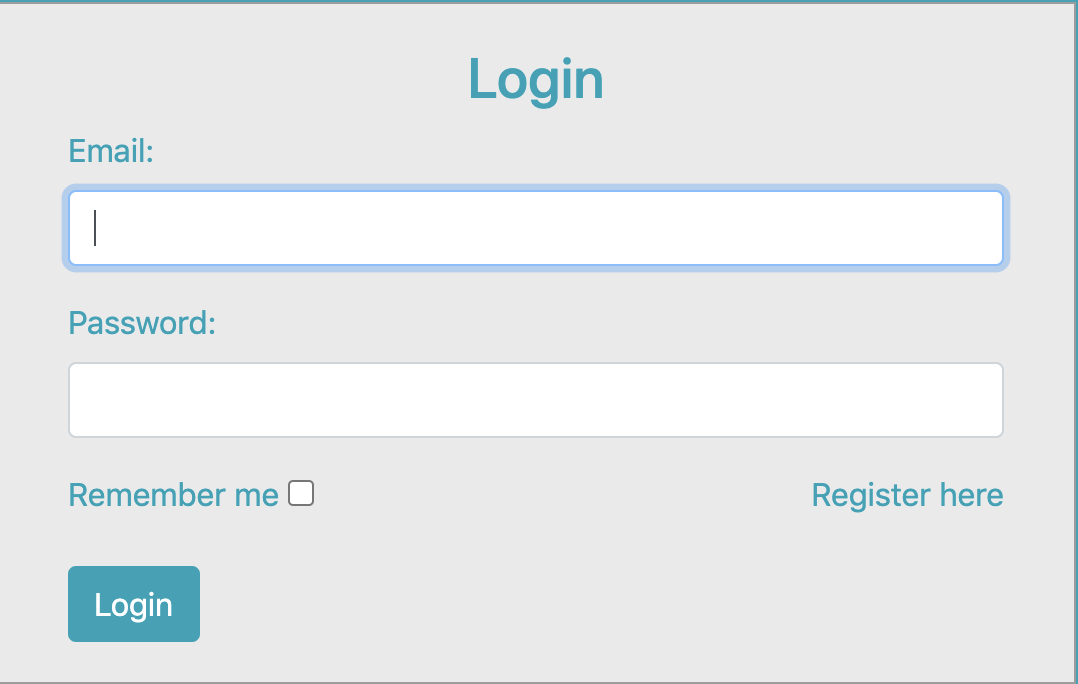


Figure : Đăng nhập hệ thống

Khi ấn vào đăng kí sẽ hiển thị thông tin để đăng kí như sau

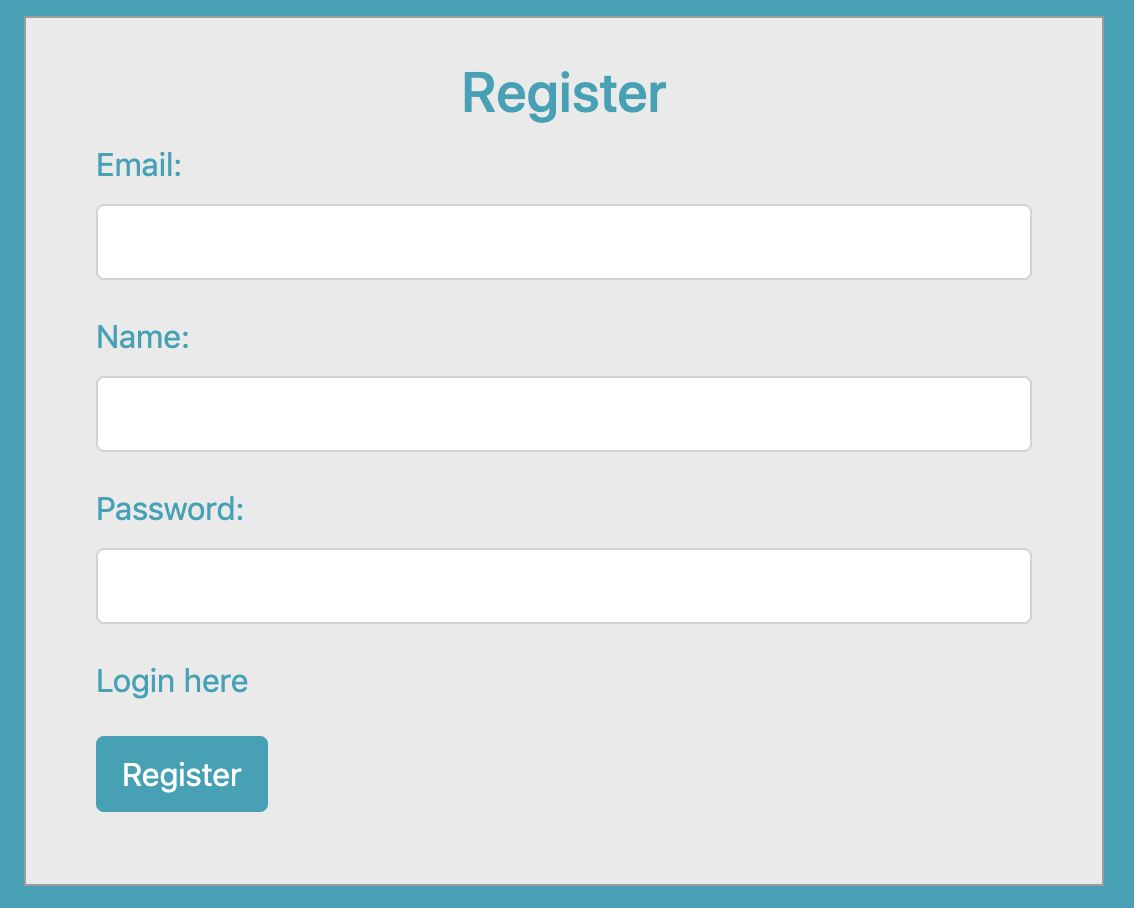


Figure : Đăng kí thành viên

## Bài tập 4: Sử dụng Laravel kết hợp với views đã học ở bài trước, thiết kế giao diện cho trang Posts

Thêm một menu nav có tên là Posts ở trên menu bar này, sau đó tạo trang CRUD cho trang tạo bài viết, ví dụ giao diện sẽ như sau

Chú ý danh sách các posts được sắp xếp theo thứ tự giảm dần của updated\_at, sử dụng orderBy(‘field’, ‘sorting\_mode’) để thực hiện việc sắp xếp này

Chú ý: Để làm việc với time GMT + 7 trong Laravel, chuyển timezone từ **UTC** -> **Asia/Ho\_Chi\_Minh** như sau để nhận giá trị timezone hiện tại của Hà Nội

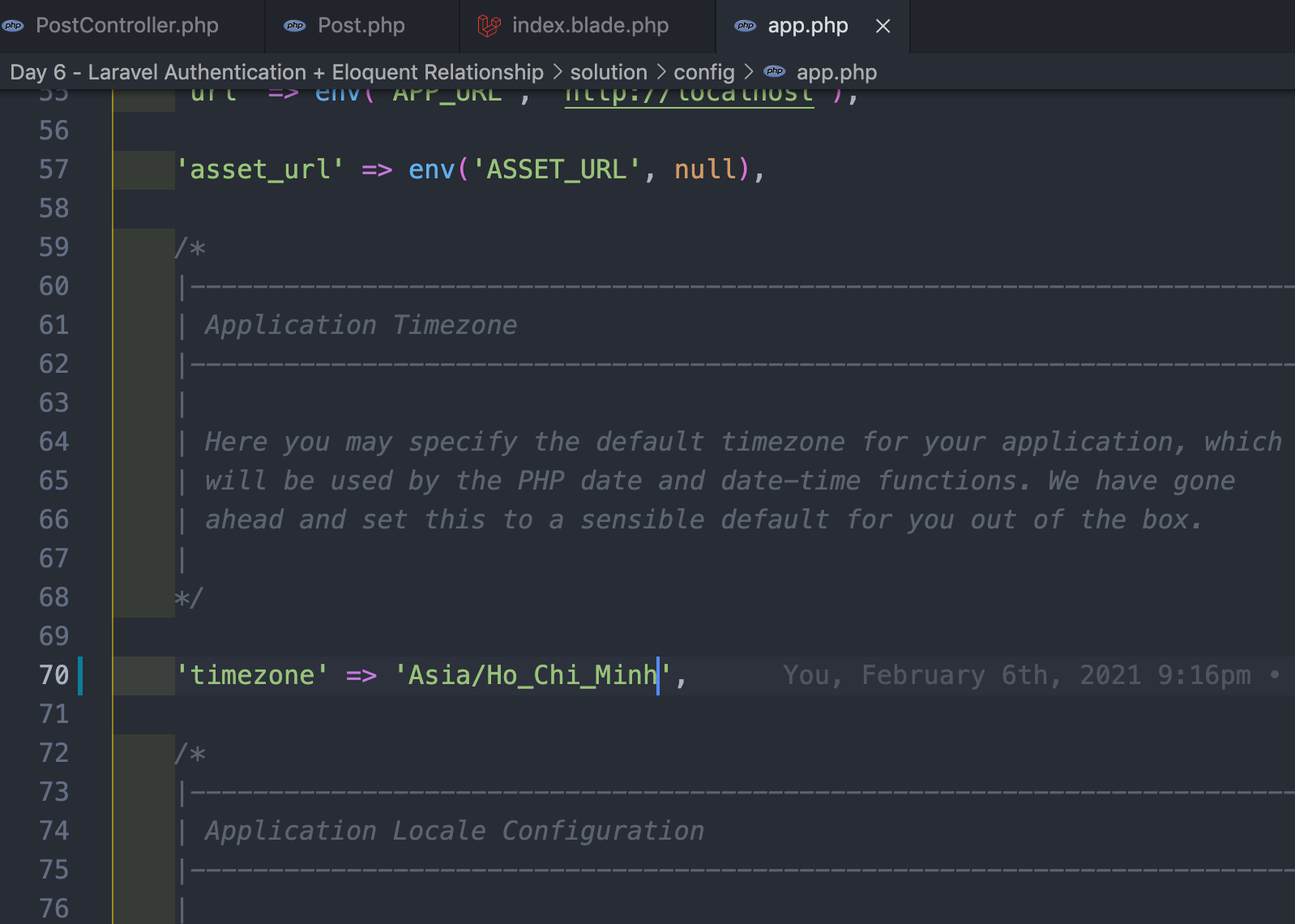


Figure : Chuyển múi giờ sang Ho Chi Minh

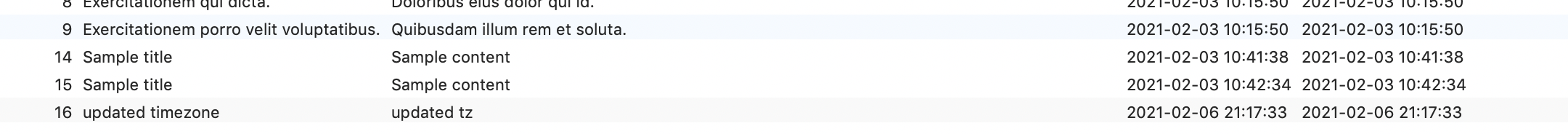


Figure : Sau khi tạo post với timezone HCM

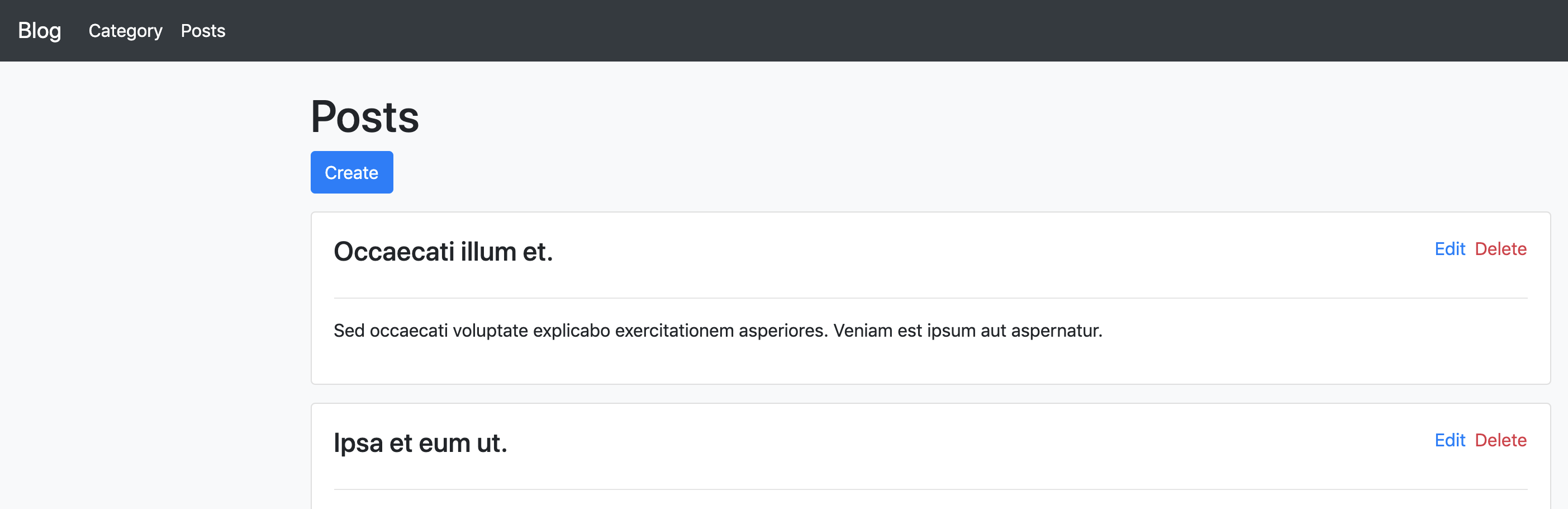


Figure : Navbar này cần có thêm posts như thế này

Đối với button “Create”, thay vì làm như trong “Category”, chúng ta sẽ sử dụng modal để tiến hành thêm thông tin cho posts mới vào CSDL. Ví dụ như sau

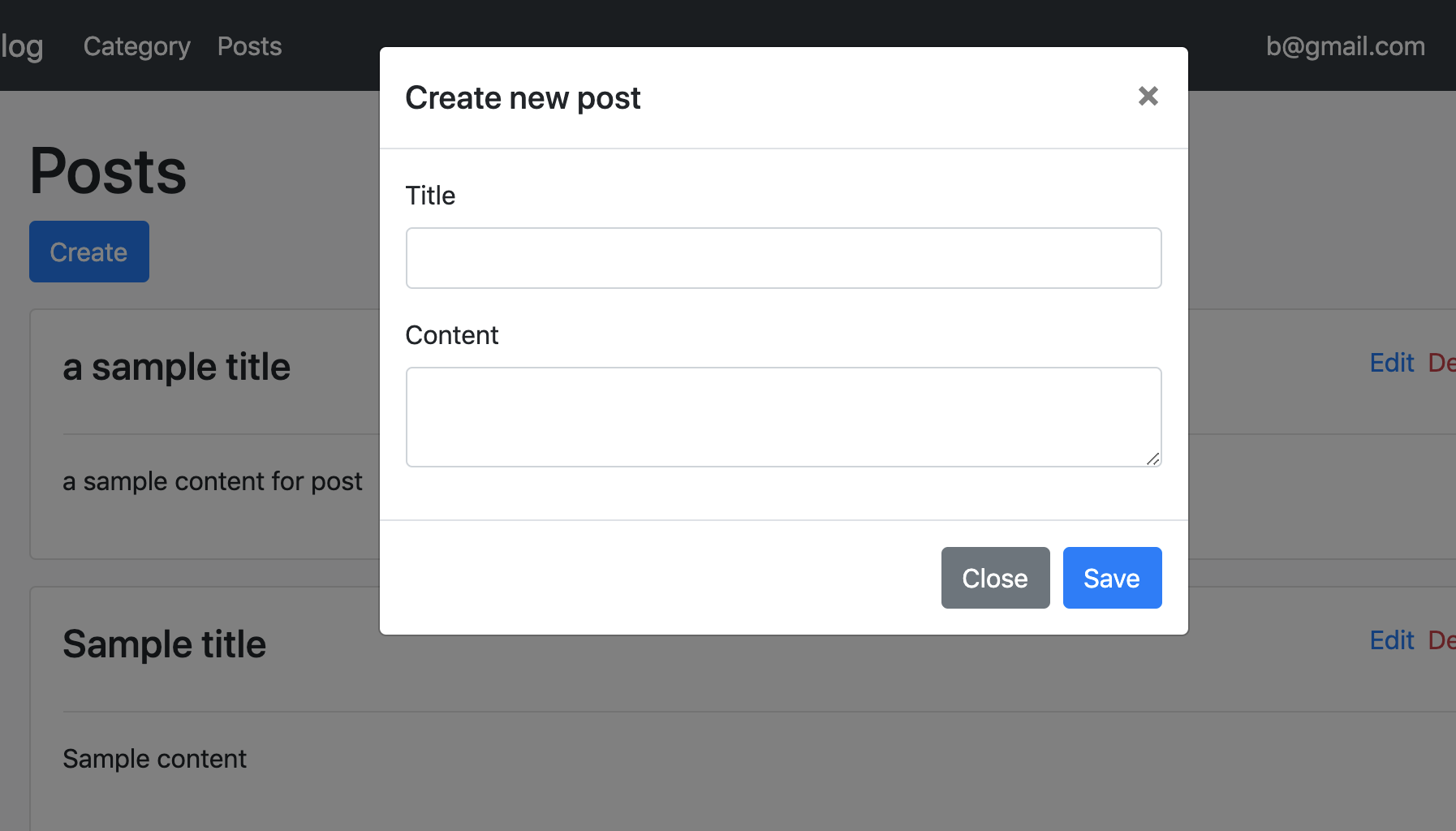


Figure : Create post thông qua modal

Để làm với modal, bọc toàn bộ modal trong form như sau



Figure : Bọc modal trong Form

Tương tự với các nút Edit và Delete như đã làm với Category. Edit sẽ tạo ra trang mới, còn Delete sẽ xoá Category đi.

## Bài tập 4: Sử dụng Eloquent Relationship để phát triển bài toán trên như sau

Hiển thị các category cho từng bài một, ví dụ như giao diện sau. Để tạo dữ liệu mẫu, có thể vào thẳng CSDL và thêm record ứng với khoá ngoại tương ứng.

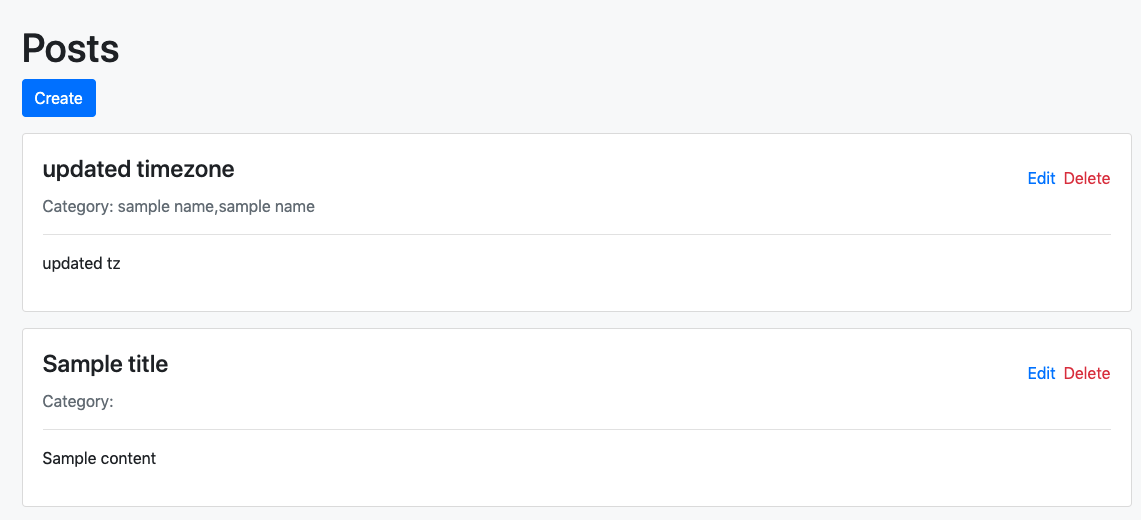


Figure : thêm category cho post

Để hiển thị được category như thế này, trong phần view chúng ta dùng hàm pluck(“field\_cần\_lấy”), kết quả trả về là Collection

Ví dụ như sau



Figure : Sử dụng eager loading để tối ưu truy vấn

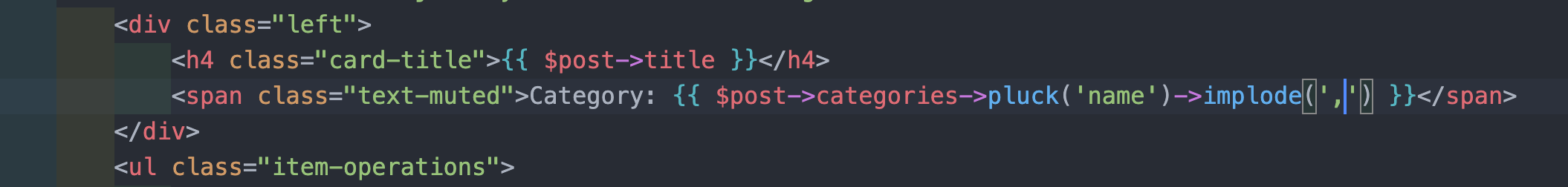


Figure : Gọi relationship và tiến hành pluck trường => trả về collection và implode

## Bài tập 5: Sử dụng kết quả của bài tập trên, thêm mới giao diện cho phép người dùng thêm category vào post trên Laravel

Ví dụ giao diện sẽ như sau (Thêm nút **Assign Categories**)

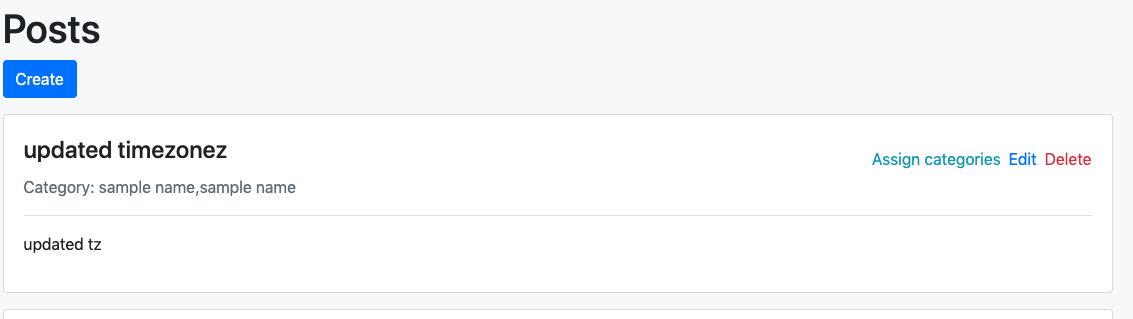


Figure : Assign category vào một post nhất định

Khi ấn vào nút “Assign categories” thì ra giao diện như sau

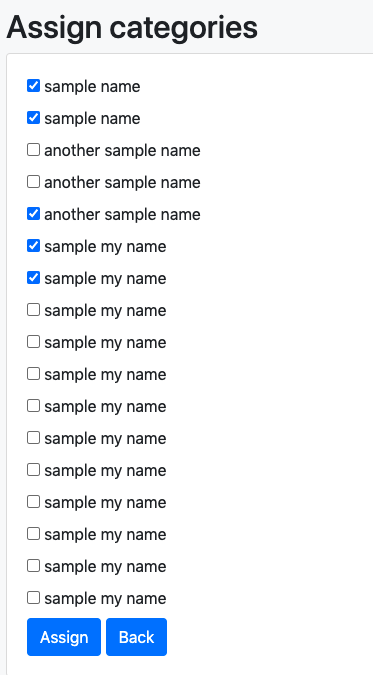


Figure : Giao diện Assign Categories

Khi ấn nút “**Assign**” thì chỉ những checkbox mà được check như hình trên thì sẽ có records ở trong bảng **posts\_categories**, chi tiết như sau

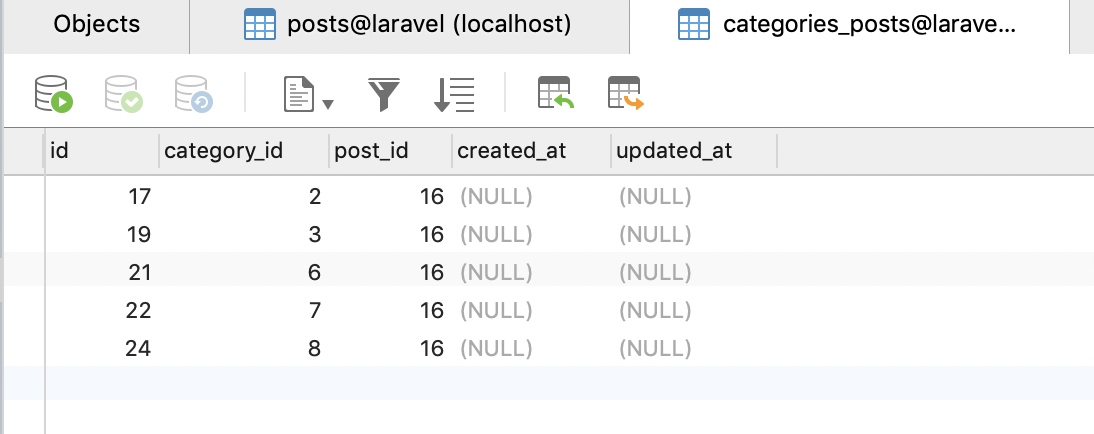


Figure : Checked items trong CSDL

Gợi ý: Để xử lí phần này thì chúng ta dùng hàm sync() trong M-M relationship

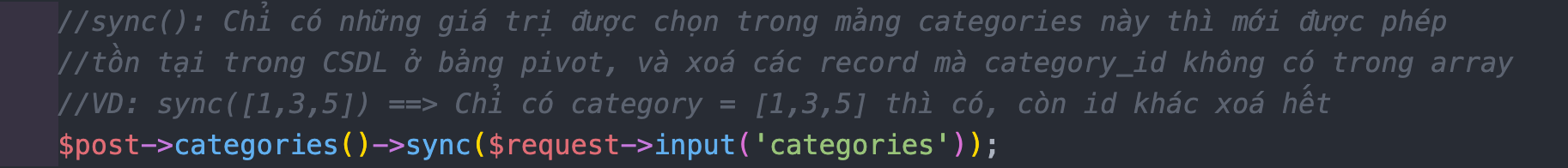


Figure : Sử dụng hàm sync() trong Laravel

## Bài tập 6: Phát triển chức năng “Comments” như đã làm ở trên mô hình đầu tiên