 TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TPHCM

KHOA CÔNG NGHỆ THÔNG TIN

\*\*\*\*\*

**ĐỒ ÁN CHUYÊN NGÀNH**

**WEBSITE CHIA SẺ Ý TƯỞNG – MVC 6**

Ngành: **CÔNG NGHỆ THÔNG TIN**

Chuyên ngành: **CÔNG NGHỆ PHẦN MỀM**

GVHD: Ths. Nguyễn Kim Hưng

SVTH: HUỲNH TUẤN PHÁT

MSSV: 1311060942 LỚP: 13DTHC04

TP. Hồ Chí Minh, 2016

Mục lục

[CHƯƠNG 1: TỔNG QUAN 6](#_Toc468046934)

[1.1. Tổng quan C# và ASP.NET MVC. 6](#_Toc468046935)

[1.1.1. Tổng quan về C#. 6](#_Toc468046936)

[1.1.2. Tổng quan về MVC ASP.NET. 7](#_Toc468046937)

[1.2. Nhiệm vụ đồ án. 11](#_Toc468046938)

[1.2.1. Phân tích hiện trạng. 11](#_Toc468046939)

[1.2.2. Mục tiêu nghiên cứu. 11](#_Toc468046940)

[1.2.3. Yêu cầu. 11](#_Toc468046941)

[CHƯƠNG 2: CƠ SỞ LÍ THUYẾT 13](#_Toc468046942)

[2.1. Công nghệ phát triển. 13](#_Toc468046943)

[2.1.1. ASP.NET Core và MVC 6. 13](#_Toc468046944)

[2.1.2. AngularJS. 14](#_Toc468046945)

[2.1.3. SingalR. 16](#_Toc468046946)

[2.1.4. Bootstrap và jQuery. 16](#_Toc468046947)

[2.2. Mô hình hóa yêu cầu (Use-case). 18](#_Toc468046948)

[2.3. Phân tích. 18](#_Toc468046949)

[2.3.1. Các thực thể. 18](#_Toc468046950)

[2.3.2. Các mối kết hợp. 19](#_Toc468046951)

[2.3.3. Mô tả chi tiết các thực thể và các mối kết hợp. 20](#_Toc468046952)

[2.3.4. Các mô hình DFD. 26](#_Toc468046953)

[2.3.5. Mô hình ERD (CDM). 32](#_Toc468046954)

[2.4. Thiết kế. 33](#_Toc468046955)

[2.4.1. Thiết kế dữ liệu (mô hình vật lí). 33](#_Toc468046956)

[2.4.2. Thiết kế giao diện. 34](#_Toc468046957)

[CHƯƠNG 3: KẾT QUẢ THỰC NGHIỆM 40](#_Toc468046958)

[3.1. Nội dung đề tài ban đầu đề ra. 40](#_Toc468046959)

[3.2. Kết quả thực hiện. 40](#_Toc468046960)

[3.2. Demo. 41](#_Toc468046961)

[CHƯƠNG 4: KẾT LUẬN 50](#_Toc468046962)

[CHƯƠNG 5: HƯỚNG PHÁT TRIỂN 51](#_Toc468046963)

[TÀI LIỆU THAM KHẢO 52](#_Toc468046964)

Danh mục hình ảnh

[Demo 1. Màn hình đăng nhập 41](#_Toc468047457)

[Demo 2. Màn hình đăng kí. 41](#_Toc468047458)

[Demo 3. Màn hình trang chủ. 42](#_Toc468047459)

[Demo 4. Màn hình đăng bài. 42](#_Toc468047460)

[Demo 5. Màn hình bài đăng. 43](#_Toc468047461)

[Demo 6. Màn hình trang chủ cùng sidebar. 43](#_Toc468047462)

[Demo 7. Màn hình chat 1. 44](#_Toc468047463)

[Demo 8. Màn hình chat 2. 45](#_Toc468047464)

[Demo 9. Màn hình quản lí tài khoản người dùng. 46](#_Toc468047465)

[Demo 10. Màn hình hiển thị danh sách nhóm. 46](#_Toc468047466)

[Demo 11. Màn hình tạo nhóm. 47](#_Toc468047467)

[Demo 12. Màn hình quản lí nhóm và chat nhóm. 47](#_Toc468047468)

[Demo 13. Màn hình quản lý bài đăng của Admin. 48](#_Toc468047469)

[Demo 14. Màn hình quản lí tải khoản người dùng của Admin. 48](#_Toc468047470)

[Demo 15. Màn hình quản lí danh mục của Admin. 49](#_Toc468047471)

LỜI CẢM ƠN

Trên thực tế không có sự thành công nào mà không gắn liền với những sự hỗ trợ, giúp đỡ dù ít hay nhiều, dù trực tiếp hay gián tiếp của người khác. Trong suốt thời gian từ khi bắt đầu học tập ở giảng đường đại học đến nay, em đã nhận được rất nhiều sự quan tâm, giúp đỡ của quý Thầy Cô, gia đình và bạn bè. Với lòng biết ơn sâu sắc nhất, em xin gửi đến quý Thầy Cô ở Khoa Công Nghệ Thông Tin – Trường Đại Học Công Nghệ TP.HCM đã cùng với tri thức và tâm huyết của mình để truyền đạt vốn kiến thức quý báu cho chúng em trong suốt thời gian học tập tại trường. Và đặc biệt, trong học kỳ này, Khoa đã tổ chức cho chúng em được tiếp cận với môn học mà theo em là rất hữu ích đối với sinh viên ngành Công Nghệ Thông Tin cũng như tất cả các sinh viên thuộc các chuyên ngành Khoa Học Kĩ Thuật khác. Em xin chân thành cảm ơn Ths. Nguyễn Kim Hưng đã tận tâm hướng dẫn. Nếu không có những lời hướng dẫn, dạy bảo của thầy thì em nghĩ bài đồ án chuyên ngành này của em rất khó có thể hoàn thiện được. Một lần nữa, em xin chân thành cảm ơn thầy.

Huỳnh Tuấn Phát

# CHƯƠNG 1: TỔNG QUAN

## 1.1. Tổng quan C# và ASP.NET MVC.

### 1.1.1. Tổng quan về C#.

#### 1.1.1.1. Ngôn ngữ C#.

C# là một ngôn ngữ lập trình đơn giản, hiện đại, mục đích tổng quát, hướng đối tượng được phát triển bởi Microsoft và được phê chuẩn bởi European Computer Manufacturers Association (ECMA) và International Standards Organization (ISO).

C# được phát triển bởi Anders Hejlsberg và team của ông trong khi phát triển .Net Framework.

C# được thiết kế cho Common Language Infrastructure (CLI), mà gồm Executable Code và Runtime Environment, cho phép chúng ta sử dụng các ngôn ngữ high-level đa dạng trên các nền tảng và cấu trúc máy tính khác nhau.

Dưới đây là các lý do làm C# là ngôn ngữ lập trình chuyên nghiệp được sử dụng rộng rãi:

* Nó là một ngôn ngữ lập trình hiện đại, mục đích tổng quát.
* Nó là hướng đối tượng.
* Nó dễ dàng để học.
* Nó là một ngôn ngữ được cấu trúc.
* Nó tạo các chương trình hiệu quả.
* Nó có thể được biên dịch trên nhiều nền tảng máy tính khác nhau.
* Nó là một phần của .Net Framework.

#### 1.1.1.2. Tại sao nên sử dụng C#?

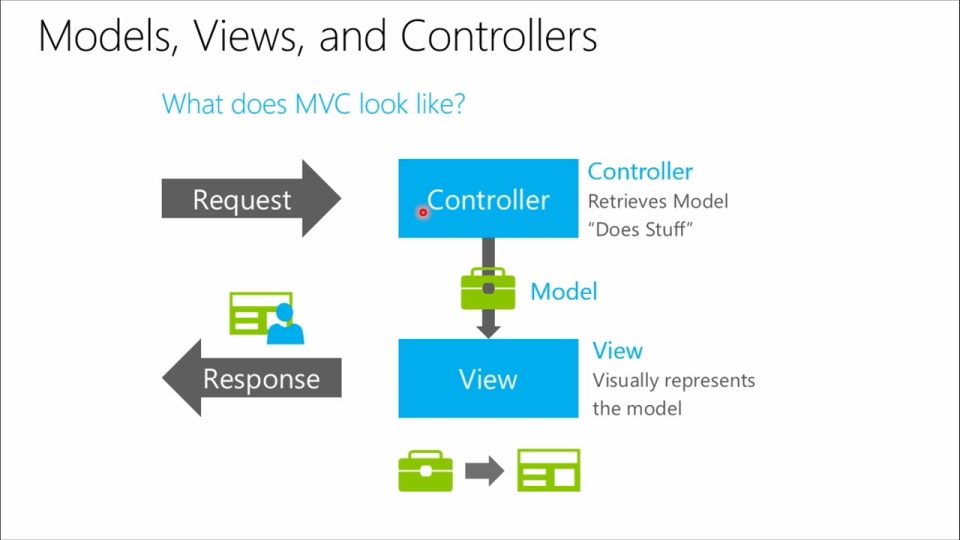
Nhiều người tin rằng không cần thiết có một ngôn ngữ lập trình mới. Java, C++, Perl, Microsoft Visual Basic, và những ngôn ngữ khác được nghĩ rằng đã cung cấp tất cả những chức năng cần thiết.

Ngôn ngữ C# là một ngôn ngữ được dẫn xuất từ C và C++, nhưng nó được tạo từ nền tảng phát triển hơn. Microsoft bắt đầu với công việc trong C và C++ và thêm vào những đặc tính mới để làm cho ngôn ngữ này dễ sử dụng hơn. Nhiều trong số những đặc tính này khá giống với những đặc tính có trong ngôn ngữ Java. Không dừng lại ở đó, Microsoft đưa ra một số mục đích khi xây dựng ngôn ngữ này. Những mục đích này được được tóm tắt như sau:

* C# là ngôn ngữ đơn giản.
* C# là ngôn ngữ hiện đại.
* C# là ngôn ngữ hướng đối tượng.
* C# là ngôn ngữ mạnh mẽ và mềm dẻo.
* C# là ngôn ngữ có ít từ khóa.
* C# là ngôn ngữ hướng module.
* C# sẽ trở nên phổ biến.

### 1.1.2. Tổng quan về MVC ASP.NET.

#### 1.1.2.1. MVC là gì?



MVC là viết tắt của Model – View – Controller. Là một kiến trúc phần mềm hay mô hình thiết kế được sử dụng trong kỹ thuật phần mềm. Nói cho dễ hiểu, nó là mô hình phân bố source code thành 3 phần, mỗi thành phần có một nhiệm vụ riêng biệt và độc lập với các thành phần khác.

#### 1.1.2.2. MVC ASP.NET.

[ASP.NET MVC](https://techmaster.vn/khoa-hoc/25503/xay-dung-mang-xa-hoi-chia-se-anh-voi-aspnet-mvc) là một framework tuyệt vời hỗ trợ pattern MVC cho ASP.NET. Nếu bạn muốn hiểu ASP.NET MVC làm việc như thế nào, bạn cần phải có một sự hiểu biết rõ ràng về mô hình MVC. MVC là cụm từ viết tắt của Model-View-Controller, nó phân chia pattern của ứng dụng thành 3 phần - model, controller và view.

Model giúp lưu trữ dữ liệu của ứng dụng và trạng thái của nó. Nó là một cơ sở dữ liệu hoặc cũng có thể chỉ là một tập tin XML.

View được coi là một giao diện người dùng được sử dụng bởi khách truy cập trang web của bạn để nhìn thấy các dữ liệu. Các trang ASPX thường được sử dụng để hiển thị view trong các ứng dụng ASP.NET MVC.

Controller chịu trách nhiệm xử lý các tương tác của người dùng với trang web. Nó được sử dụng để xác định loại view nào cần phải được hiển thị. Controller cũng được sử dụng cho mục đích giao tiếp với model.

Framework này là khá nhẹ và cung cấp khả năng kiểm thử, trong đó tích hợp với các tính năng hiện có của ASP.NET như xác thực (authentication) dựa trên membership và cả các master page.

#### 1.1.2.3. Tại sao nên sử dụng MVC ASP.NET?

Bạn nên sử dụng ASP.NET MVC framework cho việc tạo ra các ứng dụng web vì những lý do được liệt kê sau đây:

**Separation of Concern**

Một trong những lợi ích chính được đưa ra bởi ASP.NET MVC đó là Separation of Concern (phân tách mối bận tâm). Framework này cung cấp sự phân tách rất rõ gàng giữa model, logic nghiệp vụ, giao diện người dùng và dữ liệu.

**Control**

Khi bạn sử dụng framework ASP.NET MVC, bạn được cung cấp một bộ control rất mạnh mẽ trên JavaScript, HTML và CSS so với các control được cung cấp bởi một số hình thức truyền thống trên web.

**Có khả năng kiểm thử**

Framework ASP.NET MVC hỗ trợ việc kiểm thử các ứng dụng web rất tốt. Vì vậy, nếu bạn muốn kiểm thử ứng dụng web của mình trước khi đưa chúng cho người sử dụng, bạn có thể chỉ cần sử dụng ASP.NET MVC.

**Gọn nhẹ**

Framework này không sử dụng View State, hỗ trợ bạn trong việc làm giảm băng thông của các request rất nhiều. Nhiều chuyên gia khuyên bạn nên sử dụng framework này nếu bạn quan tâm đến việc tạo ra các ứng dụng MVC tuyệt vời và hữu ích dựa trên web.

**View và Size của Control**

Các framework ASP.NET thường phải đối mặt với vấn đề của view state và kích thước của control trong HTML. Phần view lưu toàn bộ dữ liệu đã được rendered và do đó kích thước các tập tin HTML trở nên lớn hơn. Nếu bạn sử dụng một đường kết nối internet chậm, bạn sẽ quan sát thấy độ trễ trong việc tải các trang web và ứng dụng. Vấn đề này không còn tồn tại trong framework ASP.NET MVC cũng như nó không chứa một khái niệm về view ở trong đó. Bằng cách sử dụng ASP.NET MVC, bạn sẽ không còn phải đối mặt với các vấn đề về thời gian tải trang nữa.

**Tích hợp**

Khi bạn tích hợp MVC với jQuery, bạn có thể viết code của mình chạy trong các trình duyệt web. Điều này sẽ giúp giảm tải cho các web server của bạn.

**Khả năng sử dụng**

Bạn không cần phải có nhiều kiến ​​thức kỹ thuật khi sử dụng framework ASP.NET MVC. Tuy nhiên, bạn sẽ thích nó sau khi học được cách làm cho nó hoạt động. Framework này tốt hơn rất nhiều so với nhiều hình thức trang web khác. Lý do là nó hoạt động khá gần với các chiến lược làm việc của web và được coi là khá dễ dàng và thú vị để sử dụng.

**API Services**

Một lợi thế lớn khác đó là việc rendered bởi MVC giúp bổ sung thêm các API web services. Nếu bạn muốn thêm các service tới ứng dụng web của mình, thì bạn chắc chắn cần phải tìm hiểu làm thế nào những service này hoạt động.

## 1.2. Nhiệm vụ đồ án.

### 1.2.1. Phân tích hiện trạng.

Nhận thấy trong quá trình làm việc, học tập, giải trí, … con người thưởng hay bị cạn ý tưởng cho những vấn đề trên. Những lúc bế tắc như thế có thể khiến bạn lâm vào những tình trạng không tốt như stress vì áp lức, tự ti về sự sáng tạo của bản thân, … Nhằm giúp người dùng khắc phục tính trạng trên nhóm đã có ý tưởng tạo ra một trang web theo hướng mạng xã hội cho người dùng giao lưu và tham khảo những ý tưởng mà người dùng khác chia sẻ.

Nhưng nói về việc chia sẻ ý tưởng, đã có một số dự án đã hoạt động theo cơ chế như trên như: Pinterest, DeviantArt, … nên nếu phát triển một dự án như vậy liệu có cạnh tranh được với những “ông lớn” trên. Vì thế, nhằm tạo ra sự khác biệt, đồng thời giúp việc thực hiện ý tưởng được dễ dàng hơn, nhóm phát triển chức năng cho phép người dùng làm việc với nhau, quản lí một dự án, …

### 1.2.2. Mục tiêu nghiên cứu.

Nhóm dự tính phát triển website này trên mô hình MVC 6 và nền tảng .NET Core được Microsoft phát triển.

Thông qua quá trình phát triển, sẽ học được nhiều hơn, cũng như tiếp cận với công nghệ mới nhằm phục vụ cho các dự án về sau.

### 1.2.3. Yêu cầu.

Danh sách các công việc thực hiện trên website (theo phân tích yêu cầu đã cho).

Chương trình cho phép:

- Quản trị viên:

1. Xem thông tin bài viết.

2. Xem thông tin người dùng (block).

3. Quản lí bài viết (xóa).

4. Quản lí chủ đề (thêm, xóa, sửa).

5. Gửi thông báo cho người dùng.

- Người dùng:

1. Quản lí thông tin cá nhân.

2. Đăng bài viết.

3. Tìm kiếm bài viết.

4. Xem bài viết.

5. Chat.

6. Bình luận, thích, chia sẻ bài viết.

7. Theo dõi người dùng khác.

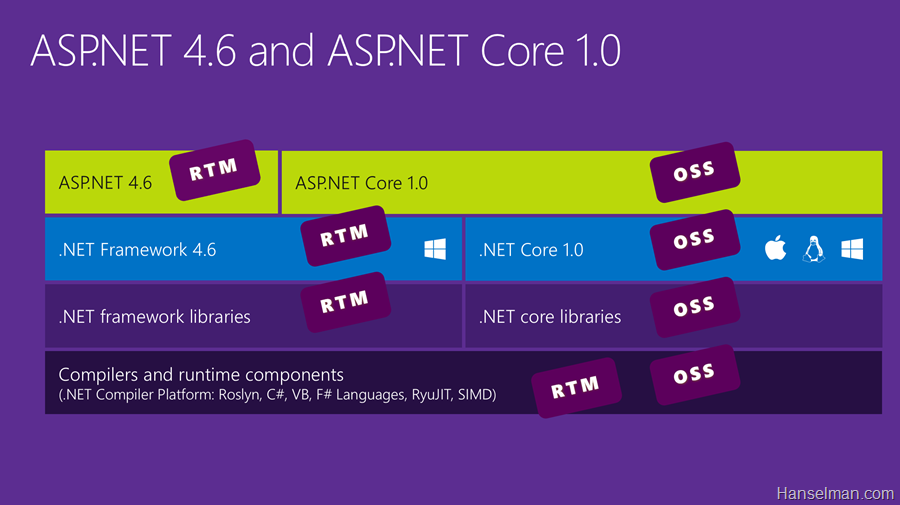
8. Tạo nhóm làm việc.

9. Chọn chủ đề yêu thích.

# CHƯƠNG 2: CƠ SỞ LÍ THUYẾT

## 2.1. Công nghệ phát triển.

### 2.1.1. ASP.NET Core và MVC 6.



ASP.NET core là một mã nguồn mở và là nền tảng mới cho xây dựng cloud trên internet kết nối các ứng dụng web, loT (Internet of Things) và mobile backends. ASP.NET Core có thể chạy trên .NET Core hoặc chạy đầy đủ trên .NET Framework. ASP.NET Core được kiến trúc để cung cấp một nền tảng phát triển tối ưu cho các ứng dụng được triển khai tới cloud hoặc chạy on-premises. ASP.NET Core bao gồm các thành phần mô-đun cần thiết tối thiểu, do đó bạn giữ lại được tính linh hoạt trong khi xây dựng các solution của bạn.

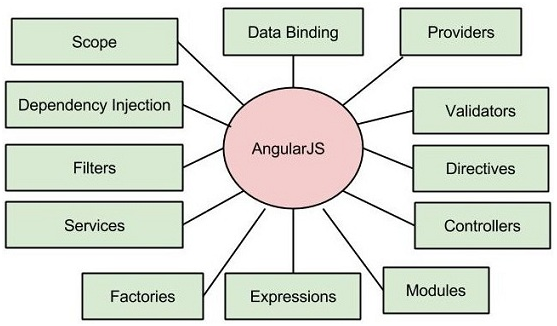
Nền tảng ASP.NET Core và MVC 6 đã được kiến trúc lại. Nhiều tính năng mới được tích hợp nổi bật ứng dụng có thể chạy trên đa nền tảng (Windows, Linux, Mac), môi trường chạy tối ưu hơn và dễ dàng cấu hình, MVC và Web Api gộp lại thành 1 framework duy nhất sử dụng chung đường ống http mới. Một số điểm mới:

* Chạy trên đa nền tảng Windows, Linux và Mac
* Một nền tảng duy nhất cho MVC, API
* Template web mặc định mới
* Hiệu năng HTTP nhanh hơn
* Hệ thống cấu hình mới
* Tích hợp sẵn Dependency Injection
* Bổ sung các tool phổ biến phía client bower, grunt, glup
* TagHelper
* View Components

### 2.1.2. AngularJS.

Angular là một bộ Javascript Framework rất mạnh và thường được sử dụng để xây dựng project **Single Page Application**(SPA). Nó hoạt động dựa trên các thuộc tính mở rộng HTML (các atributes theo quy tắc của Angular). Đây là một Framework mã nguồn mở hoàn toàn miễn phí và được hàng ngàn các lập trình viên trên thế giới ưa chuộng và sử dụng. Framework này được thế hệ Web 2.0 phát triển khá mạnh ở nước ngoài, tuy nhiên ở Việt Nam thì vẫn chưa thông dụng lắm.

* AngularJS là một Framwork phát triển dựa trên Javascript để tạo các ứng dụng web phong phú
* AngularJS thường dùng để phát triển frontend (giao diện khách hàng) thông qua các API để gọi data, sử dụng mô hình MVC rất mạnh mẽ
* Mã nguồn AngularJS tự động fix với các trình duyệt khác nhau nên bạn không cần phải lo vấn đề tương thích trình duyệt
* Angular là mã nguồn mở, hoàn toàn miễn phí và được phát triển bởi hàng ngàn các lập trình viên trên thế giới.



Các thành phần chính của AngularJS:

* **Data-binding**: (liên kết dữ liệu) tự động đồng bộ dữ liệu giữa model và view
* **Scope**: (Phạm vi) Đây là những đối tượng kết nối giữa Controller và View
* **Controller**: Đây là những hàm javascript xử lý kết hợp với bộ điều khiển Scope
* **Service**: Như tôi đề cập ở trên, AngularJS sử dụng các API được xây dựng từ các web service (PHP, ASP) để thao tác với DB.
* **Filters**: Bộ lọc lọc ra các thành phẩn của một mảng và trả về mảng mới
* **Directives**:  đánh dấu vào các yếu tố của DOM, nghĩa là sẽ tạo ra các thẻ HTML tùy chỉnh
* **Templates**: hiển thị thông tin từ controller, đây là một thành phần của views
* **Routing**:  chuyển đổi giữa các action trong controller
* **MVC**: Mô hình chia thành phần riêng biệt thành Model, View, Controller. Đây là một mô hình khá hay nhưng trong Angular thì nó được chế biến lại một chút gần giốn với MVVM (Model View View Model)
* **Deep Linking:** Liên kết sâu, cho phép bạn mã hóa trạng thái của ứng dụng trong các URL để nó có thể đánh dấu được với công cụ tìm kiếm.
* **Dependency Injection:** Angular giúp các nhà phát triển tạo ứng dụng dễ dàng hơn để phát triển, hiểu và thử nghiệm dễ dàng.

### 2.1.3. SingalR.

SignalR là 1 thư viện mã nguồn mở viết cho .NET giúp xây dựng các ứng dụng web sử dụng tương tác thời gian thực giữa người dùng với máy chủ. Ví dụ như: Mạng xã hội, games online, tin tức, ứng dụng thời tiết, bản tin tài chính, giá chứng khoán, ứng dụng chat, ...

Mô hình bình thường của web là client gửi request đến server, nhưng giờ ta có 1 cách thức nữa để tương tác: server “push” trực tiếp dữ liệu về client, theo thời gian thực. Ngoài ra, mô hình này cũng giúp ta xây dựng ứng dụng bất đồng bộ lên đến hàng nghìn kết nối 1 lúc với máy chủ.

### 2.1.4. Bootstrap và jQuery.

Bootstrap hiện đang là Framwork hỗ trợ tốt nhất và phổ biến nhất trong việc phát triễn các dự án về giao diện website cho cả thiết bị di động và máy tính bao gồm HTML, CSS và Java script. Bootstrap phát triễn qua nhiều phiên bản, phiên bản mới nhất lúc mình thực hiện tutorials là bản 3.3.2.

Bootstrap được sử dụng rộng rải bởi lợi ích to lớn mà nó mang lại như giao diện website tương thích với các thiết bị smartphone, tablet và máy tính thông thường vì có thể tự động co giản kích thước. Bootstrap được xây dựng với các module hỗ trợ sẵn nên rất dễ dàng tìm hiểu, xây dựng giao diện một cách nhanh chóng…

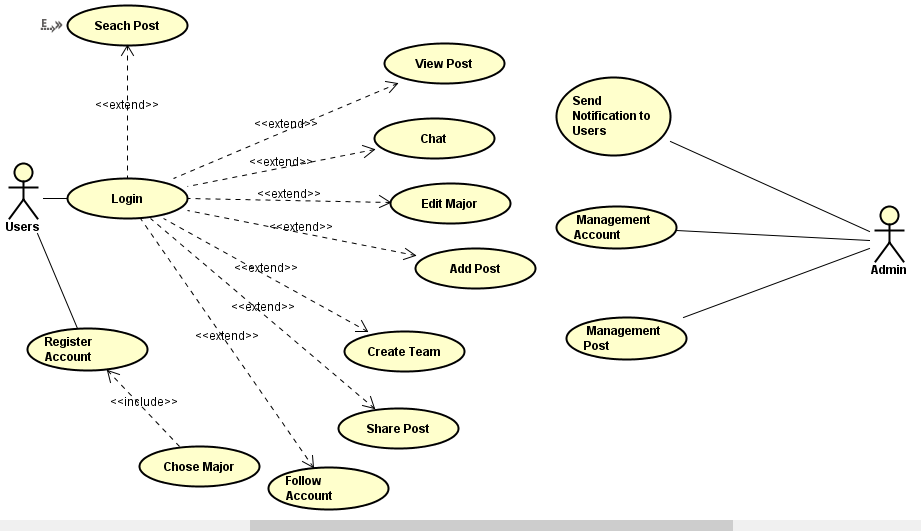
jQuery là một thư viện kiểu mới của JavaScript, được tạo bởi John Resig vào năm 2006 với một phương châm tuyệt vời: **Write less, do more**.

jQuery làm đơn giản hóa việc truyền tải HTML, xử lý sự kiện, tạo hiệu ứng động và tương tác Ajax. Với jQuery, khái niệm Rapid Web Development đã không còn quá xa lạ.

jQuery là một bộ công cụ tiện ích JavaScript làm đơn giản hóa các tác vụ đa dạng với việc viết ít code hơn. Dưới đây liệt kê một số tính năng tối quan trọng được hỗ trợ bởi jQuery:

* **Thao tác DOM −** jQuery giúp dễ dàng lựa chọn các phần tử DOM để *traverse* một cách dễ dàng như sử dụng CSS, và chỉnh sửa nội dung của chúng bởi sử dụng phương tiện Selector mã nguồn mở, mà được gọi là **Sizzle**.
* **Xử lý sự kiện −** jQuery giúp tương tác với người dùng tốt hơn bằng việc xử lý các sự kiện đa dạng mà không làm cho HTML code rối tung lên với các Event Handler.
* **Hỗ trợ AJAX −** jQuery giúp bạn rất nhiều để phát triển một site giàu tính năng và phản hồi tốt bởi sử dụng công nghệ AJAX.
* **Hiệu ứng −** jQuery đi kèm với rất nhiều các hiệu ứng đa dạng và đẹp mắt mà bạn có thể sử dụng trong các Website của mình.
* **Gọn nhẹ −** jQuery là thư viện gọn nhẹ - nó chỉ có kích cỡ khoảng 19KB (gzipped).
* **Được hỗ trợ hầu hết bởi các trình duyệt hiện đại −** jQuery được hỗ trợ hầu hết bởi các trình duyệt hiện đại, và làm việc tốt trên IE 6.0+, FF 2.0+, Safari 3.0+, Chrome và Opera 9.0+
* **Cập nhật và hỗ trợ các công nghệ mới nhất −** jQuery hỗ trợ CSS3 Selector và cú pháp XPath cơ bản.

## 2.2. Mô hình hóa yêu cầu (Use-case).



## 2.3. Phân tích.

### 2.3.1. Các thực thể.

|  |  |  |
| --- | --- | --- |
| Tên thực thể | Code | Mô tả |
| Connection | Connections | Quản lí kết nối người dùng với server. |
| Idea | Ideas | Những ý tưởng người dùng đăng lên. |
| Team | Teams | Hoạt động nhóm. |
| Type | Type | Phân loại bài đăng, nhóm. |
| UserAccount | UserAccounts | Tài khoản người dùng |
| UserSearched | UserSearched | Lịch sử tìm kiếm của người dùng. |

### 2.3.2. Các mối kết hợp.

|  |  |  |
| --- | --- | --- |
| Tên | Code | Mô tả |
| Chat | Chats | Chat giữa người dùng với nhau |
| Comment | Comments | Comment của người dùng với bài đăng |
| Follow | Follows | Người dùng theo dõi nhau. |
| IdeaLiked | IdeaLikeds | Người dùng thích bài đăng. |
| IdeaShared | IdeaShareds | Người dùng chia sẻ bài đăng. |
| IdeaType | IdeaTypes | Bài đăng thuộc nhiều danh mục. |
| InviteNotify | InviteNotifys | Lời mời của leader nhóm gửi đến người dùng. |
| TeamChat | TeamChats | Chat nhóm. |
| TeamType | TeamTypes | Team thuộc nhiều danh mục. |
| UserTeam | UserTeams | Người dùng thuộc những nhóm. |
| UserType | UserType | Danh mục yêu thích của người dùng. |

### 2.3.3. Mô tả chi tiết các thực thể và các mối kết hợp.

#### 2.3.3.1. Các thực thể.

Thực thể Ideas.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Code | Datatype | Primary Identifier | Null |
| Id | Id | Int | True | False |
| Create Day | CreateDay | Datetime | False | False |
| Description | Description | Nvarchar(100) | False | False |
| Image | Image | Varchar(500) | False | False |
| Like Count | LikeCount | Int | False | False |
| Share Count | ShareCount | Int | False | False |
| Tag | Tag | Nvarchar(50) | False | False |
| Title | Title | Nvarchar(50) | False | False |
| UserAcountId | UserAcountId | Int | False | False |

Thực thể Types.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Code | Datatype | Primary Identifier | Null |
| Id | Id | Int | True | False |
| Description | Description | Nvarchar(100) | False | False |
| Name | Name | Nvarchar(50) | False | False |

Thực thể Teams.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Code | Datatype | Primary Identifier | Null |
| Id | Id | Int | True | True |
| CreateDate | CreateDate | Datatime | False | False |
| Name | Name | Nvarchar(50) | False | False |
| Title | Tile | Nvarchar(50) | False | False |

Thực thể UserSearcheds.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Code | Datatype | Primary Identifier | Null |
| Id | Id | Int | True | False |
| Key Word | KeyWord | Nvarchar(50) | False | False |
| SearchDate | SearchDate | DateTime | False | False |

Thực thể UserAccounts.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Code | Datatype | Primary Identifier | Null |
| Id | Id | Int | True | False |
| Avatar | Avatar | Varchar(200) | False | False |
| Birthday | Birthday | Datetime | False | False |
| Email | Email | Varchar(50) | False | False |
| Full Name | FullName | Nvarchar(50) | False | False |
| Locked Account | LockedAccount | Bit | False | False |
| Password | Password | Varchar(30) | False | False |
| Phone | Phone | Numeric(18,0) | False | False |
| UserName | UserName | Varchar(50) | False | False |
| VerifyAccount | VerifyAccount | Int | False | False |

Thực thể Connections

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Code | Datatype | Primary Identifier | Null |
| Id | Id | Int | True | False |
| ConnectionId | ConnectionId | Varchar(100) | False | False |
| UserId | UserId | Int | False | False |
| Time | Time | Datetime | False | False |
| IsConnected | IsConnected | Bool | False | False |

#### 2.3.3.2. Các mối kết hợp.

TeamTypes.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Code | Datatype | Primary Identifier | Null |
| TeamId | TeamId | Int | True | False |
| TypeId | TypeId | Int | True | False |

UserTeams.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Code | Datatype | Primary Identifier | Null |
| TeamId | TeamId | Int | True | False |
| UserId | UserId | Int | True | False |
| Position | Position | Nvarchar(50) | False | False |

IdeaTypes.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Code | Datatype | Primary Identifier | Null |
| IdeaId | IdeaId | Int | True | False |
| TypeId | TypeId | Int | True | False |

IdeaShareds.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Code | Datatype | Primary Identifier | Null |
| IdeaId | IdeaId | Int | True | False |
| UserAccountId | UserAccountId | Int | True | False |
| ShareDate | ShareDate | DateTine | False | False |

IdeaLikeds.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Code | Datatype | Primary Identifier | Null |
| IdeaId | IdeaId | Int | True | False |
| UserAccount | UserAccount | Int | True | False |

Follows.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Code | Datatype | Primary Identifier | Null |
| Flwerld | Flwerd | Int | True | False |
| Flwingld | Flwingld | Int | True | False |

Comments.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Code | Datatype | Primary Identifier | Null |
| IdeaId | IdeaId | Int | True | False |
| UserAccount | UserAccount | Int | True | False |
| CommentTime | CommentTime | DateTime | False | False |
| Content | Content | Nvarchar(200) | False | False |

Chats

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Code | Datatype | Primary Identifier | Null |
| UserFromId | UserFromId | Int | True | False |
| UserToId | UserToId | Int | True | False |
| SendDate | SendDate | DateTime | True | False |
| Message | Message | nvarchar(300) | False | True |
| Seen | Seen | Bool | False | False |

InviteNotifys

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Code | Datatype | Primary Identifier | Null |
| FromId | FromId | Int | True | False |
| ToId | ToId | Int | True | False |
| TeamId | TeamId | Int | True | False |
| Status | Status | Bool | False | False |
| Date | Date | Datetime | True | False |

TeamChats

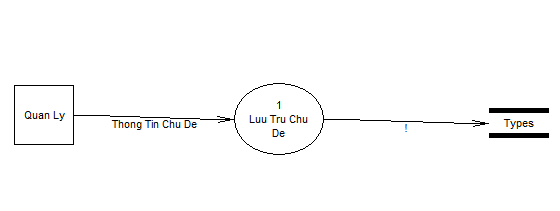
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Code | Datatype | Primary Identifier | Null |
| TeamId | TeamId | Int | True | False |
| UserId | UserId | Int | True | False |
| Message | Message | Nvarchar(300) | False | True |
| SendDate | SendDate | Datetime | True | False |

UserTypes

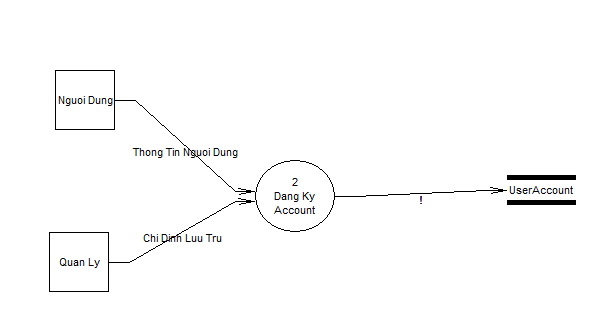
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Code | Datatype | Primary Identifier | Null |
| UserId | UserId | Int | True | False |
| TypeId | TypeId | Int | True | False |

### 2.3.4. Các mô hình DFD.

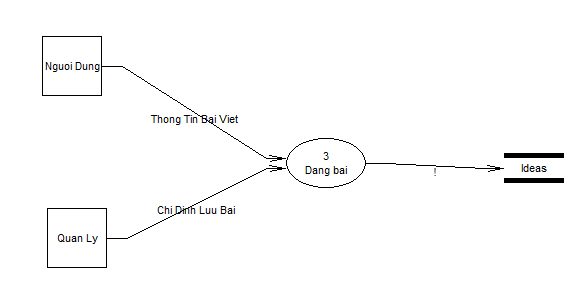
#### 2.3.4.1. Lưu Trữ Types.



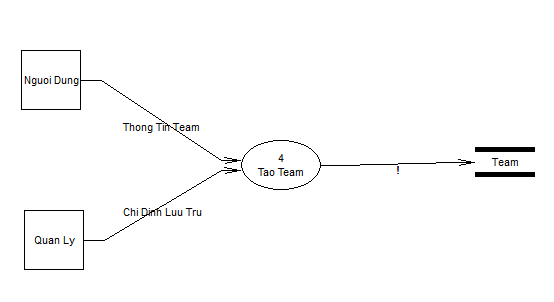
#### 2.3.4.2. Lưu Trữ UserAccounts.



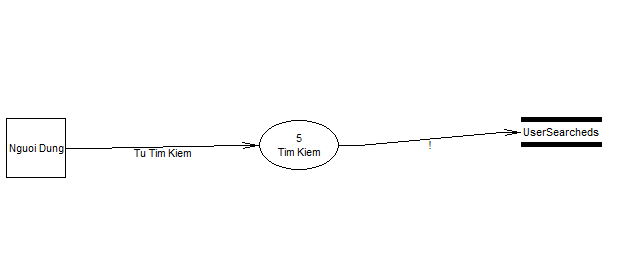
#### 2.3.4.3. Lưu Trữ Ideas.



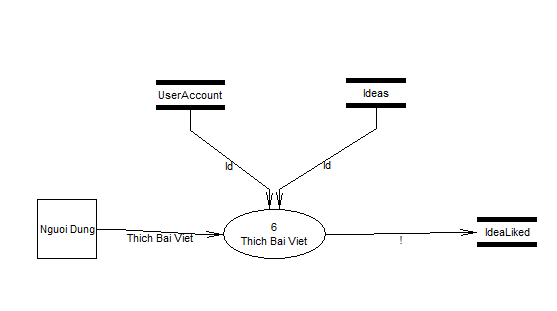
#### 2.3.4.4. Lưu Trữ Teams.



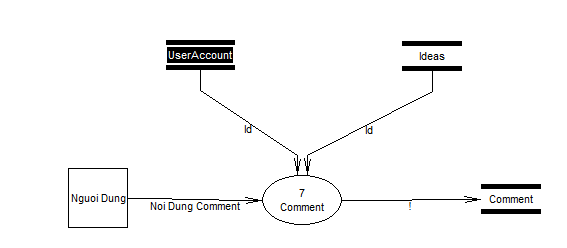
#### 2.3.4.5. Lưu Trữ Usersearcheds.



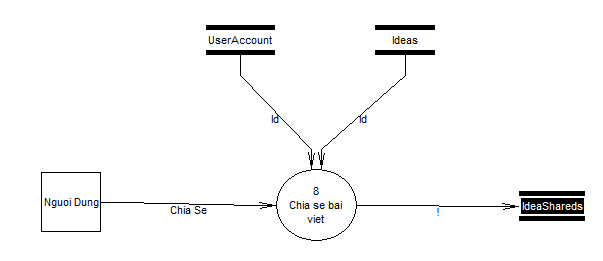
#### 2.3.4.6. Lưu Trữ IdeaLikes.



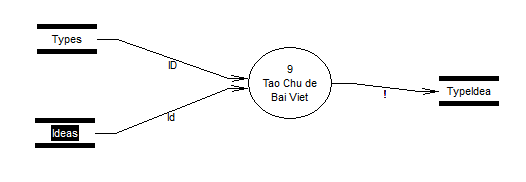
#### 2.3.4.7. Lưu Trữ Comments.



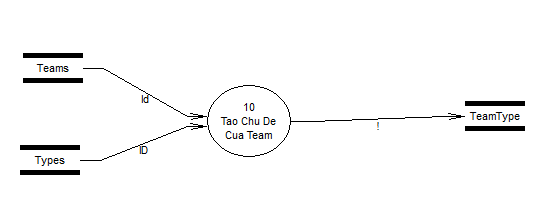
#### 2.3.4.8. Lưu Trữ IdeaShareds.



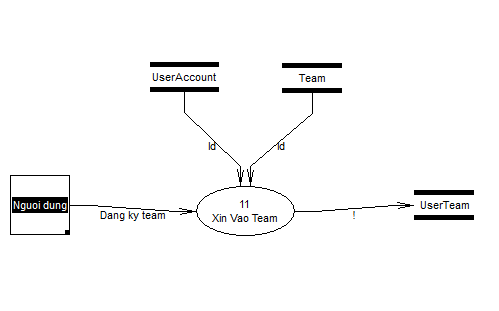
#### 2.3.4.9. Lưu Trữ TypeIdeas.



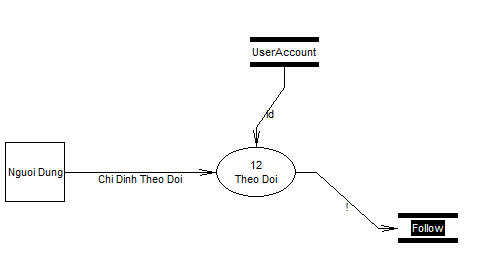
#### 2.3.4.10. Lưu Trữ TeamTypes.



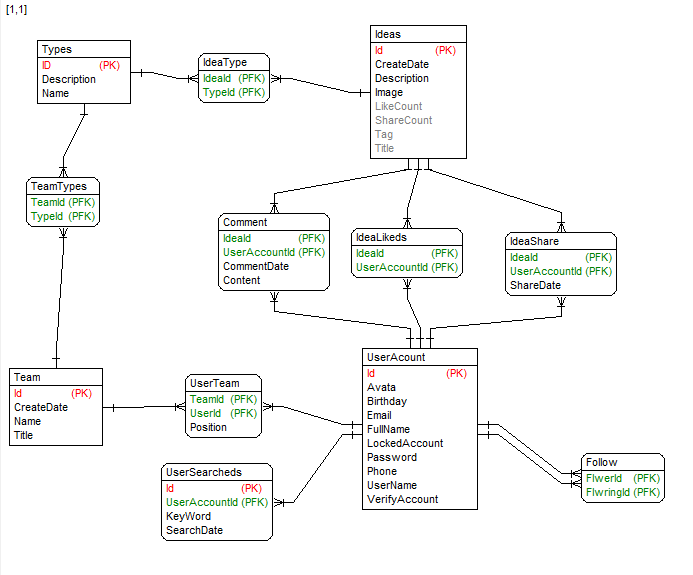
#### 2.3.4.11. Lưu Trữ UserTeams.



#### 2.3.4.12. Lưu Trữ Follows.

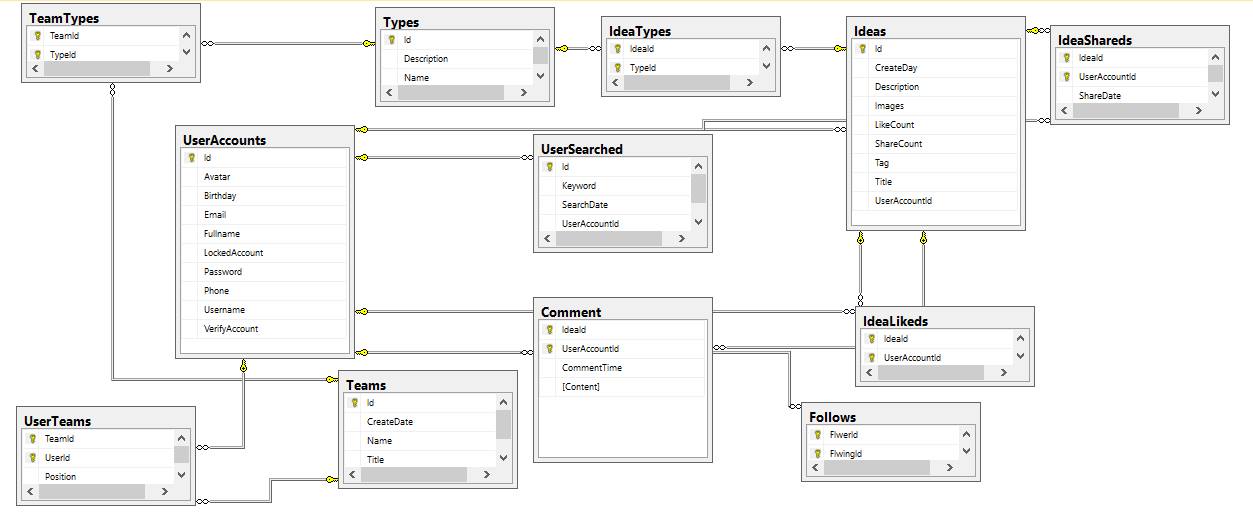


### 2.3.5. Mô hình ERD (CDM).

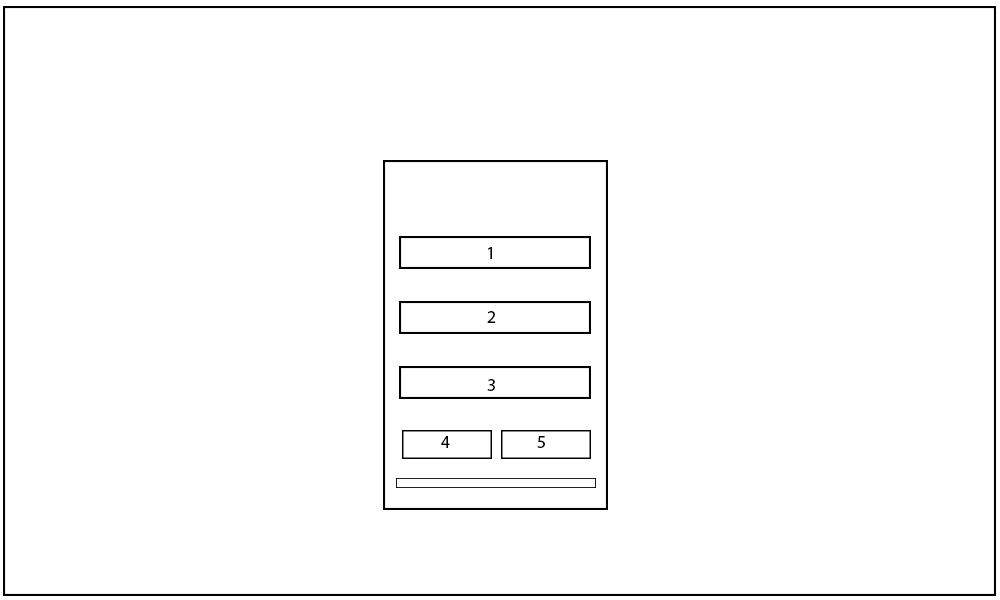


## 2.4. Thiết kế.

### 2.4.1. Thiết kế dữ liệu (mô hình vật lí).

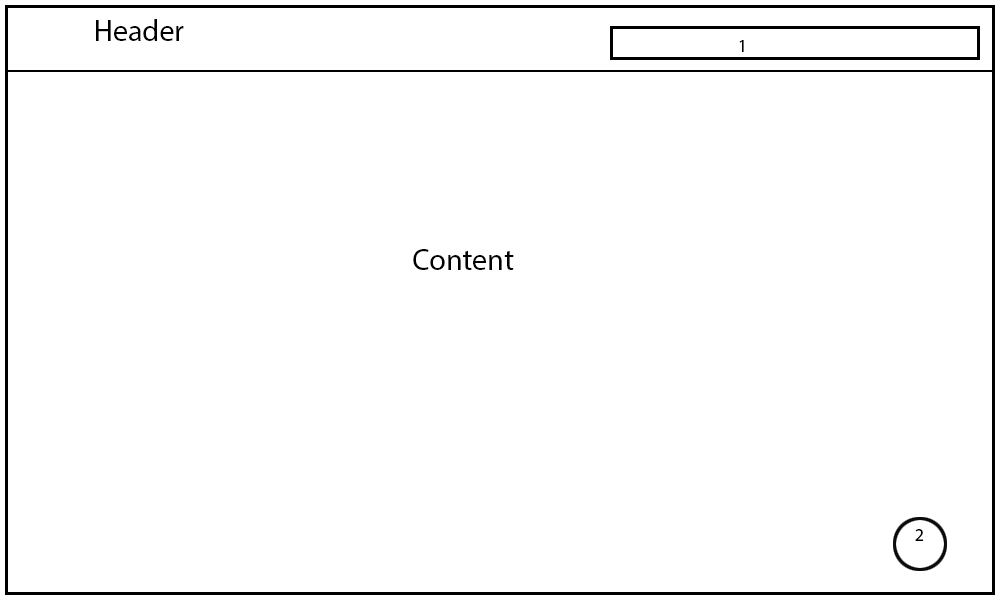


### 2.4.2. Thiết kế giao diện.



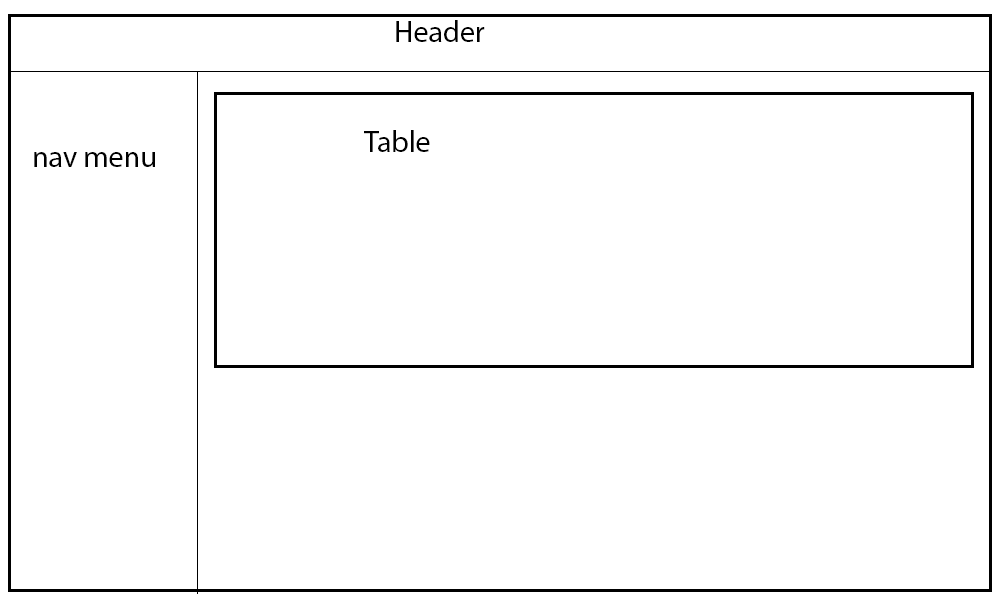
Hình . Giao diện màn hình đăng nhập.

|  |  |
| --- | --- |
| STT | Mô tả |
| 1 | Ô Điền tên tài khoản |
| 2 | Ô Điền mật khẩu |
| 3 | Button sign in |
| 4 | Login Facbook |
| 5 | Login Google |



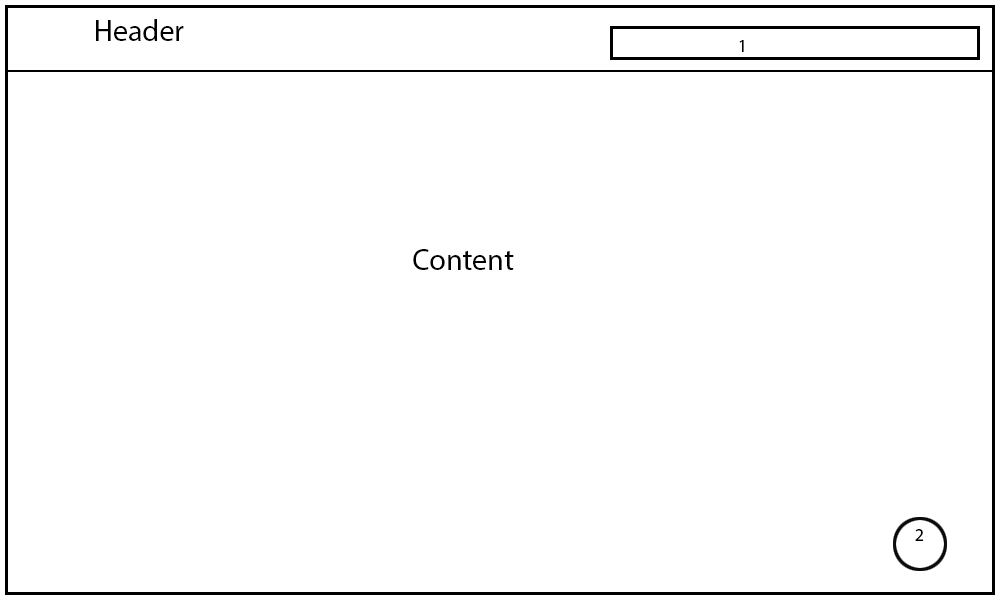
Hình . Giao diện màn hình chính.

|  |  |
| --- | --- |
| STT | Mô tả |
| 1 | Khung tìm kiếm |
| 2 | Button them bài viết |

­­­­­

Hình . Giao diện Sơ đồ doanh thu.

|  |  |
| --- | --- |
| STT | Mô tả |
| 1 | Nav menu hiện thị các chức năng quản lý |
| 2 | Table hiện thị thông tin quản lý. |

Hình . Giao diện màn hình thêm bài viết.

8

7

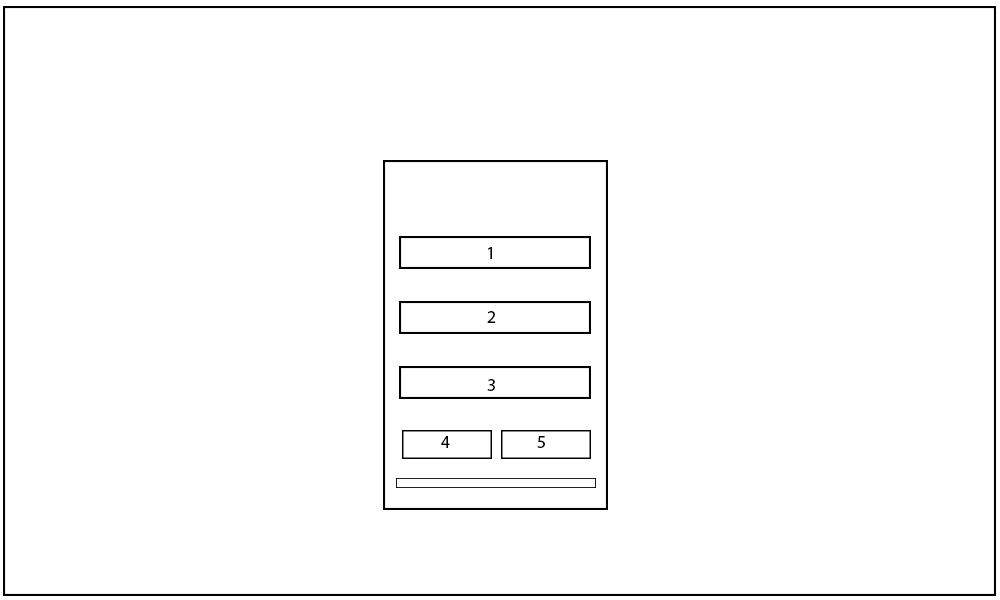
4

3

5

6

|  |  |
| --- | --- |
| STT | Mô tả |
| 1 | Khung tìm kiếm |
| 2 | Button thêm bài viết |
| 3 | Ô nhập Title |
| 4 | Ô nhập description |
| 5 | Button them hình ảnh |
| 6 | Ô nhập tags |
| 7 | Ô nhập fields |
| 8 | Button Post |



6

5

4

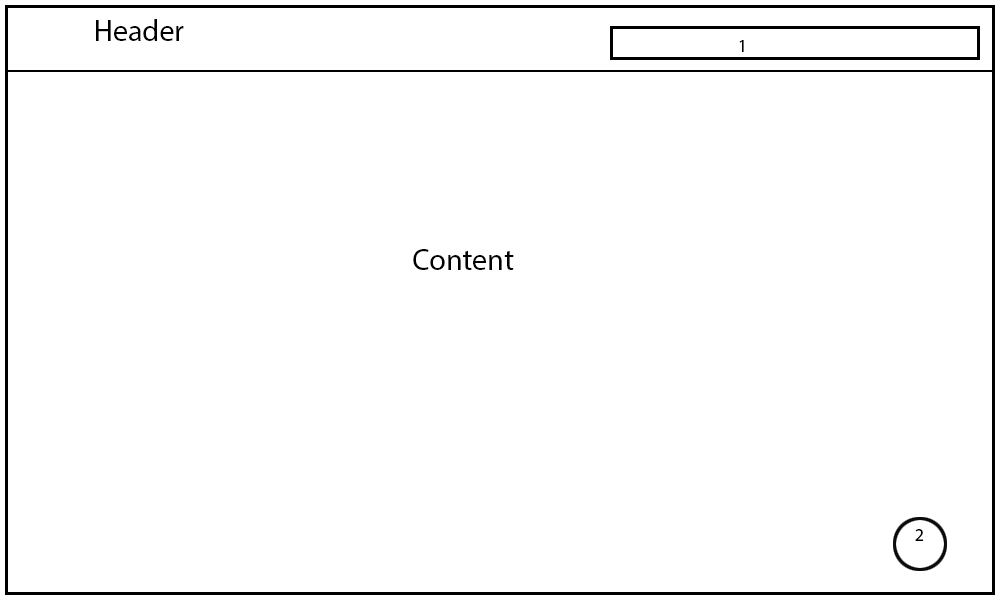
3

2

1

Hình . Giao diện màn hình đăng ký tài khoản.

|  |  |
| --- | --- |
| STT | Mô tả |
| 1 | Ô nhập Fullname |
| 2 | Ô nhập Password |
| 3 | Ô nhập Email |
| 4 | Ô nhập Username |
| 5 | Ô nhập Re-type password |
| 6 | Button sign up |



7

6

4

5

3

Hình . Giao diện màn hình đăng ký tài khoản.

|  |  |
| --- | --- |
| STT | Mô tả |
| 1 | Khung tìm kiếm |
| 2 | Button thêm bài viết |
| 3 | Hình ảnh bài viết |
| 4 | Hiên thị thông tin người dùng |
| 5 | Ô nhập comment |
| 6 | Button share |
| 7 | Button like |

# CHƯƠNG 3: KẾT QUẢ THỰC NGHIỆM

## 3.1. Nội dung đề tài ban đầu đề ra.

Đề tài ban đầu đề ra nhằm mục đích trao đổi ý tưởng giữa người dùng với nhau. Các chức năng cơ bản đã hoàn thành như đăng bài, bình luận, làm việc nhóm, … Ngoài ra nhóm đã triển khai thực tế lên host và không gặp vấn đề gì về lỗi.

## 3.2. Kết quả thực hiện.

Hoàn thành được 90% mục tiêu đề ra.

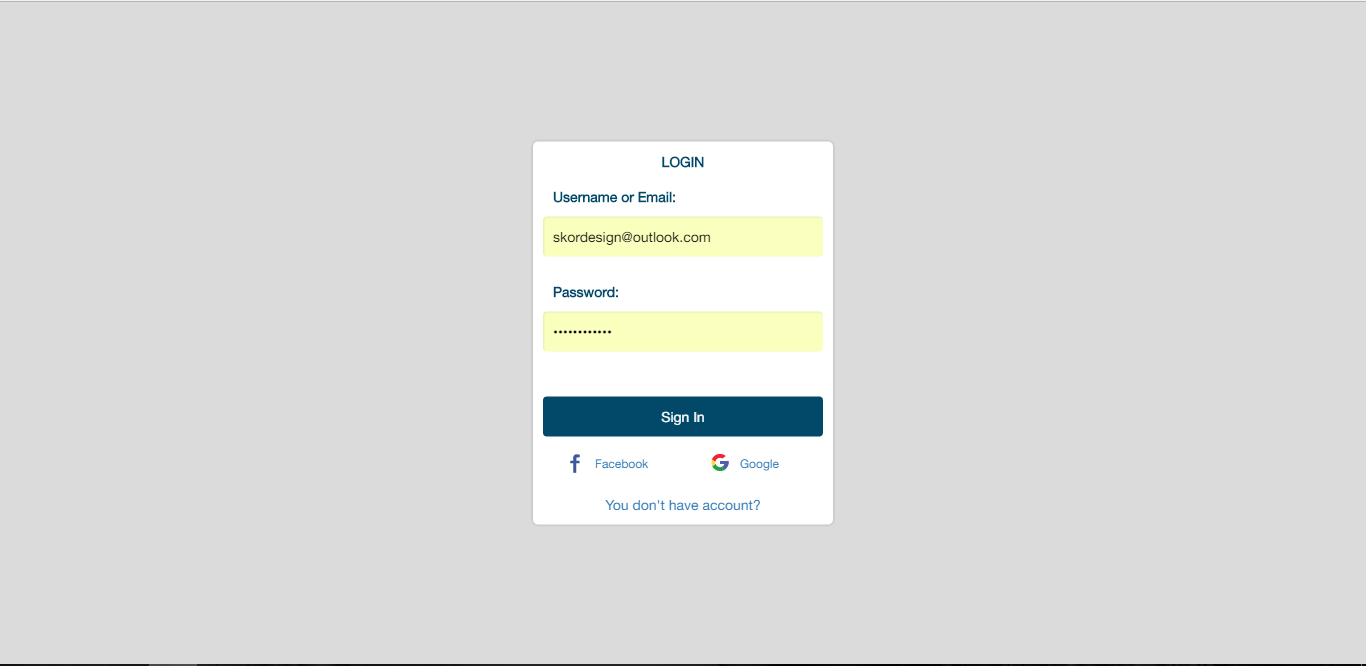
Các chức năng hoạt động khác ổn định và rất ít khi lỗi.

Trong quá trình thực hiện đề tài, nhóm đã học được cách làm một ứng dụng web ASP.NET Core trên môi trường Windows, cách thiết kế một website không quá cầu kì mà hiệu quả đối với người dùng, ngoài ra con biết thêm về sử dụng HTML, CSS và framework AngularJS trước đây chưa từng dùng đến. Nói chung qua đề tài này nhóm đã học được nhiều thứ khá mới mẻ đối với bản thân.

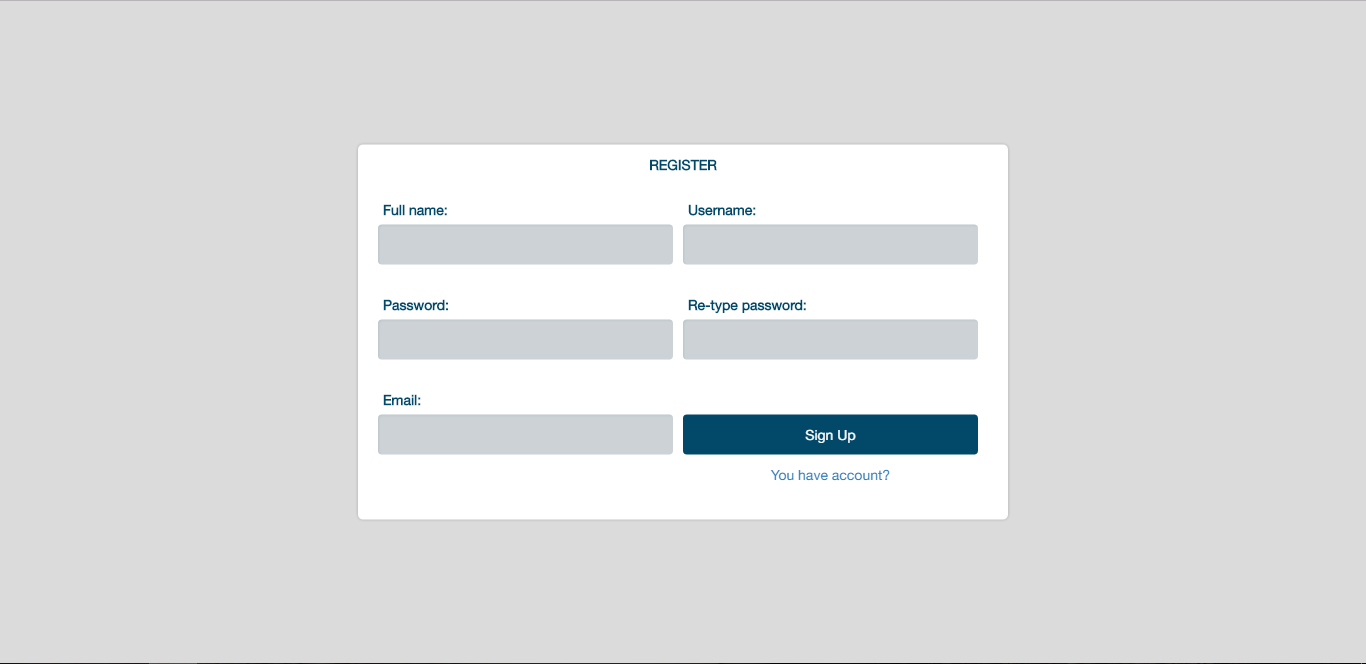
Ngoài ra, nhóm còn học được cách quản lí các static packages bằng bower và npm. Đóng gói file tính với gulp và webpack. Những công cụ này đang được cộng đồng developer nước ngoài dùng rất nhiều.

## 3.2. Demo.

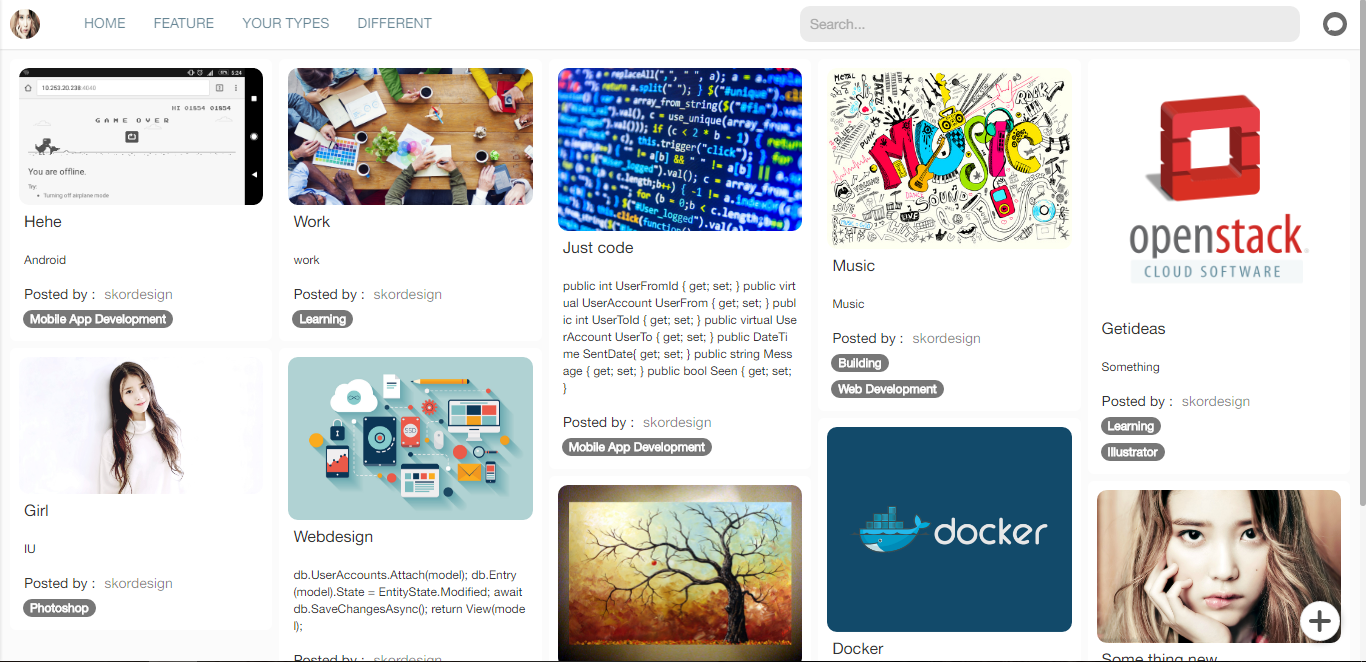
Website: http://getideas20161124121135.azurewebsites.net



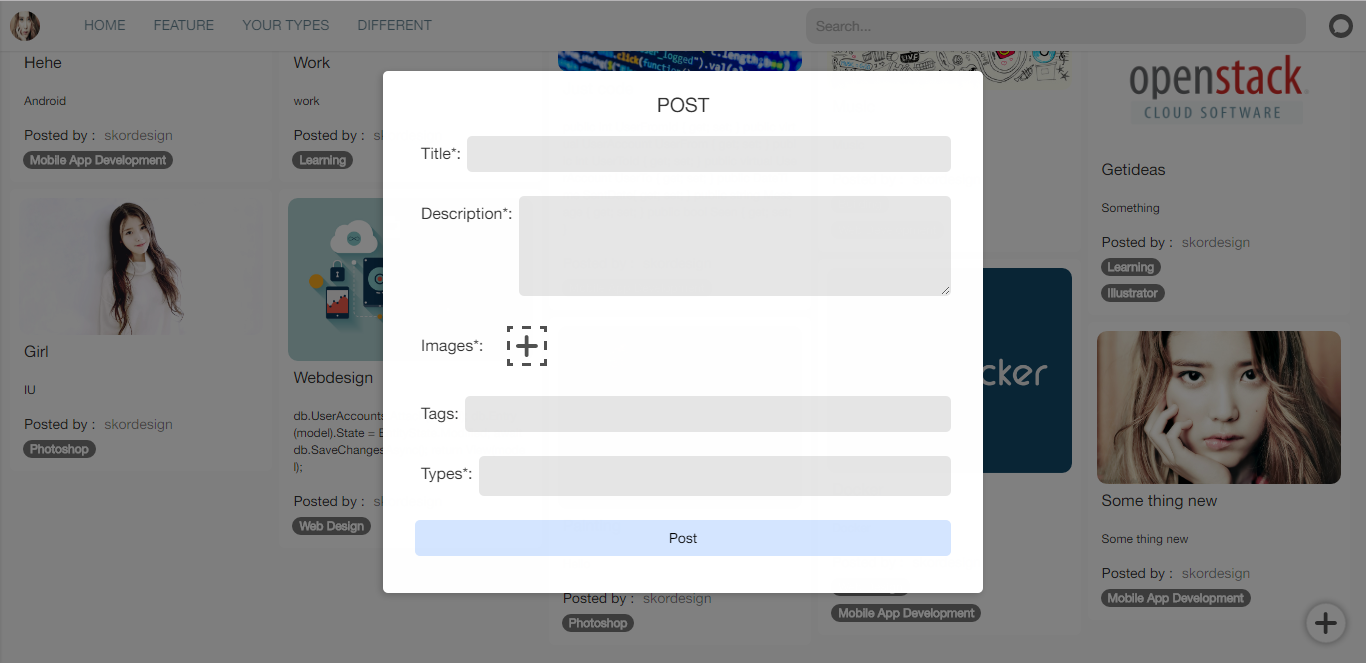
Demo . Màn hình đăng nhập



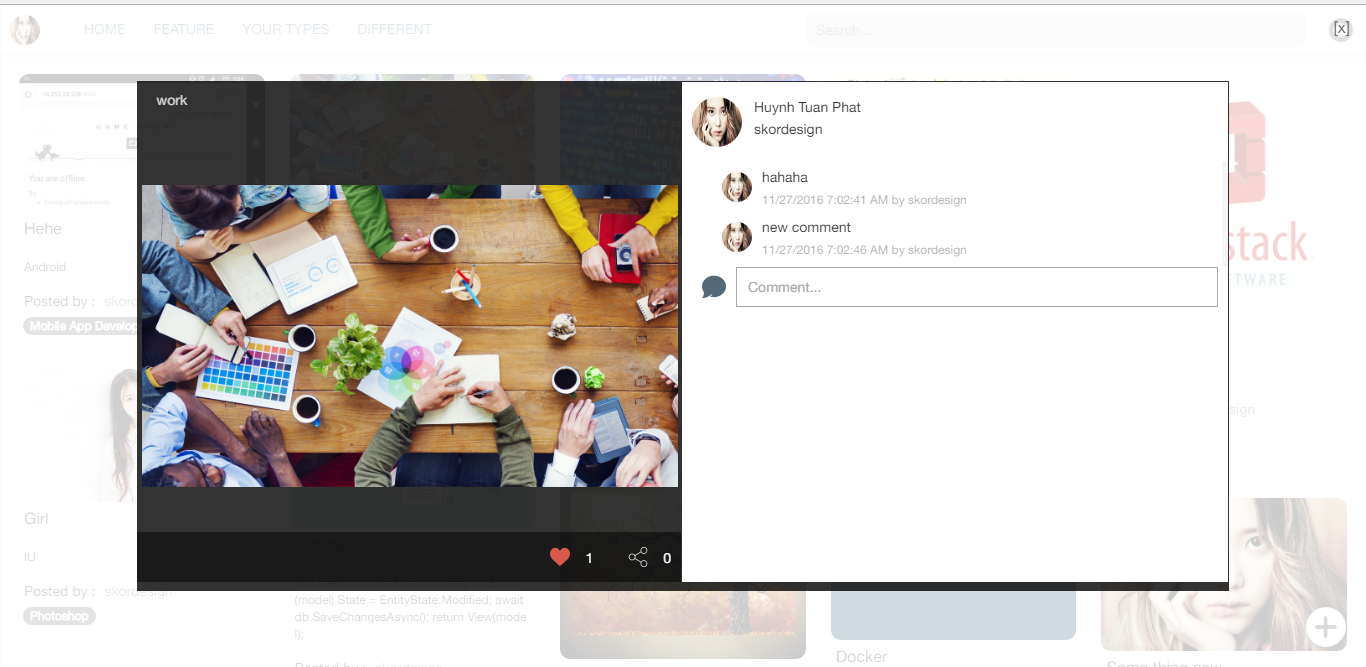
Demo . Màn hình đăng kí.



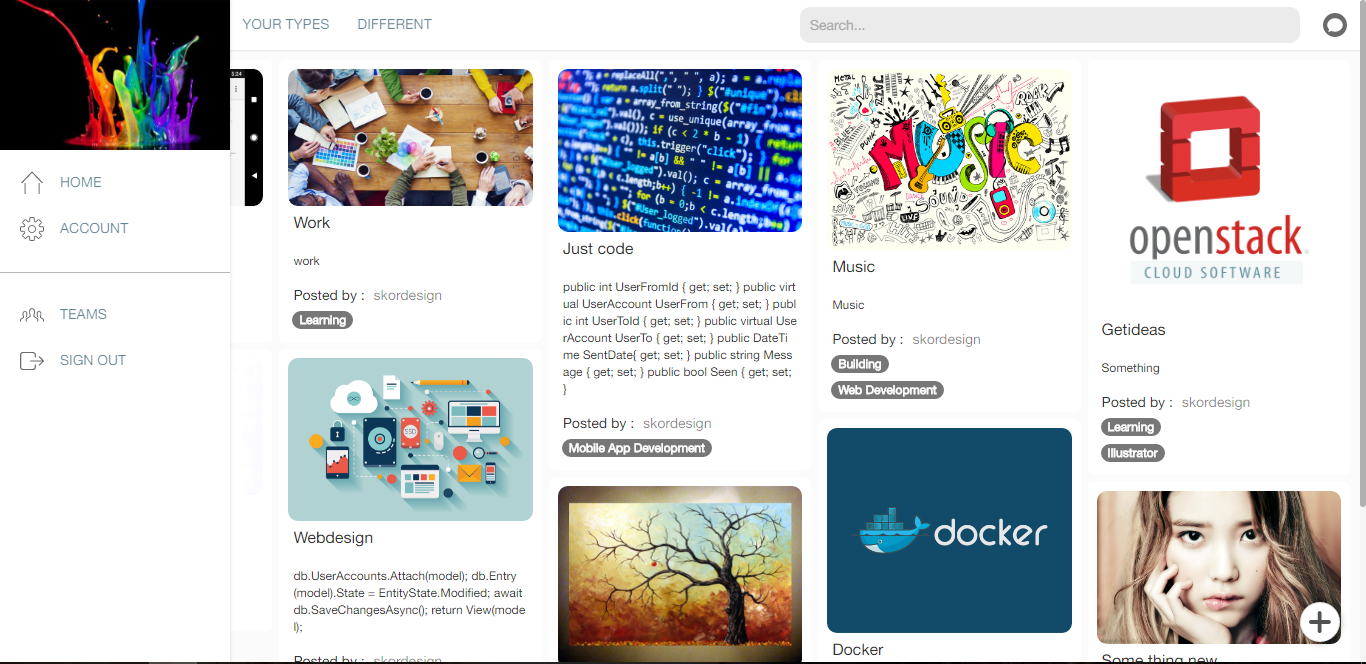
Demo . Màn hình trang chủ.



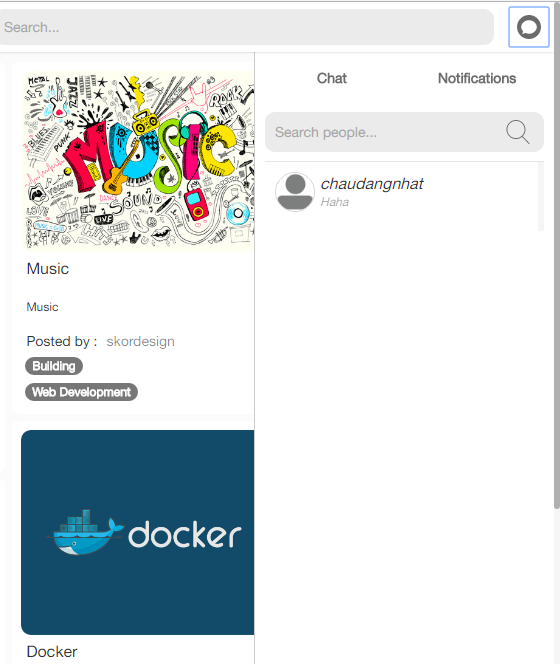
Demo . Màn hình đăng bài.



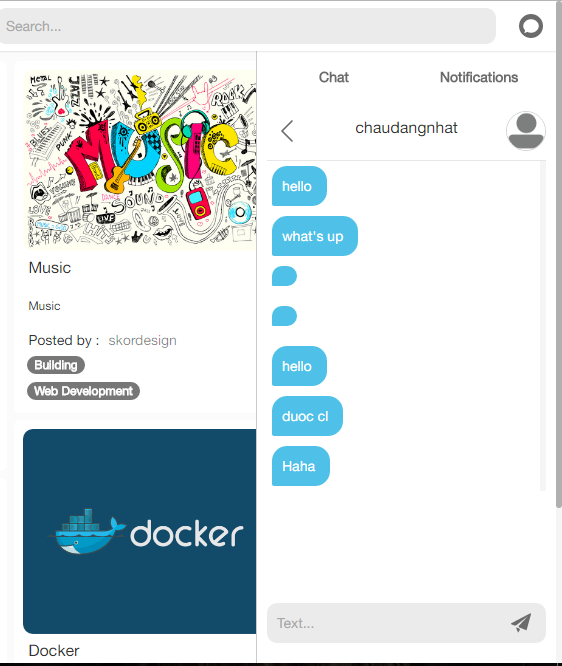
Demo . Màn hình bài đăng.



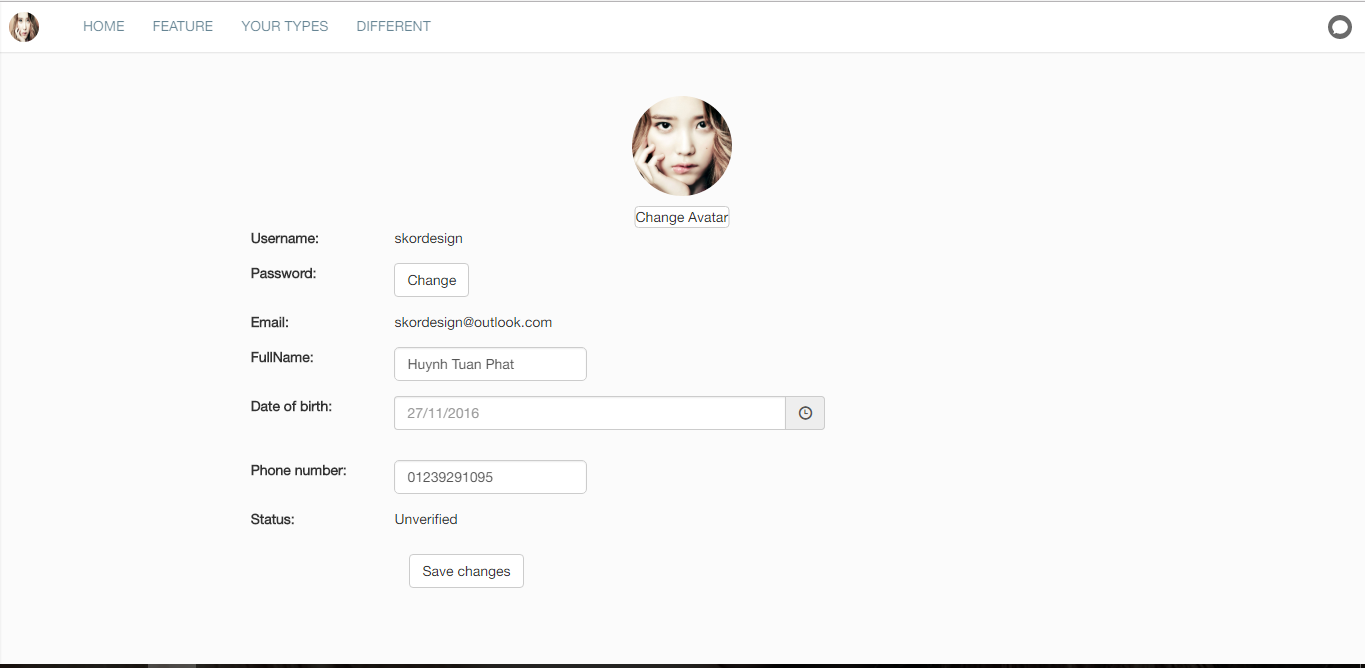
Demo . Màn hình trang chủ cùng sidebar.



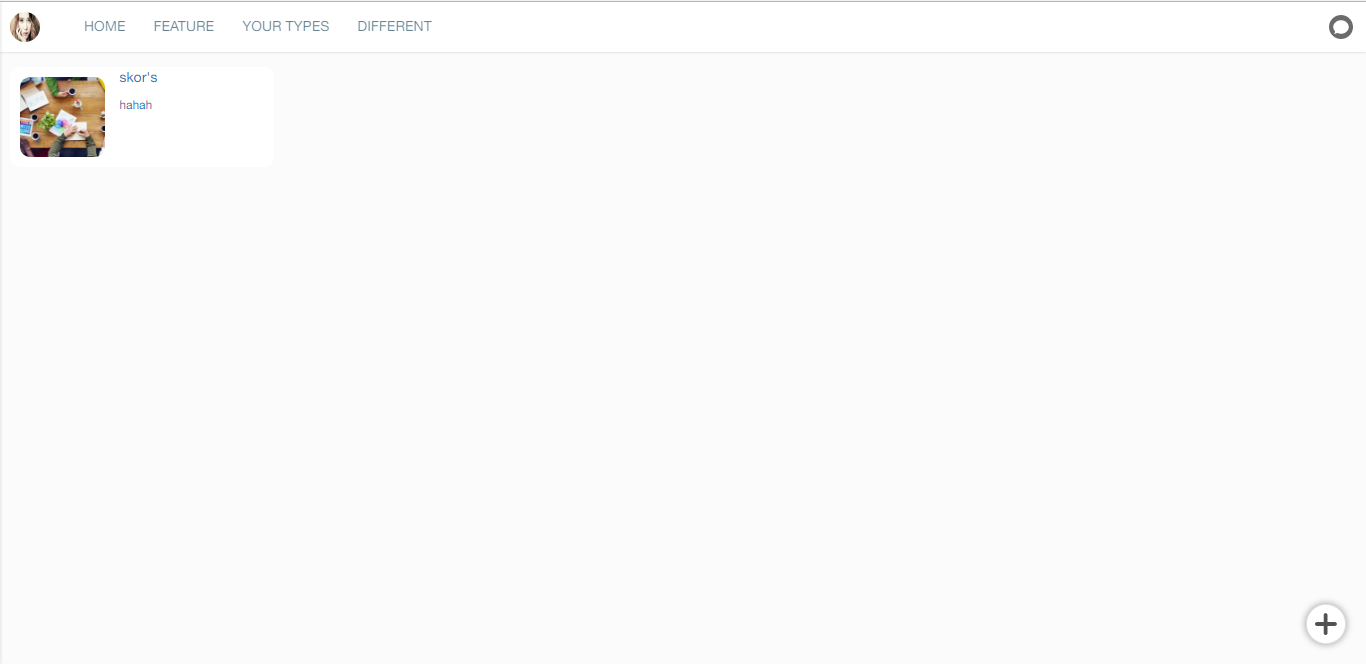
Demo . Màn hình chat 1.



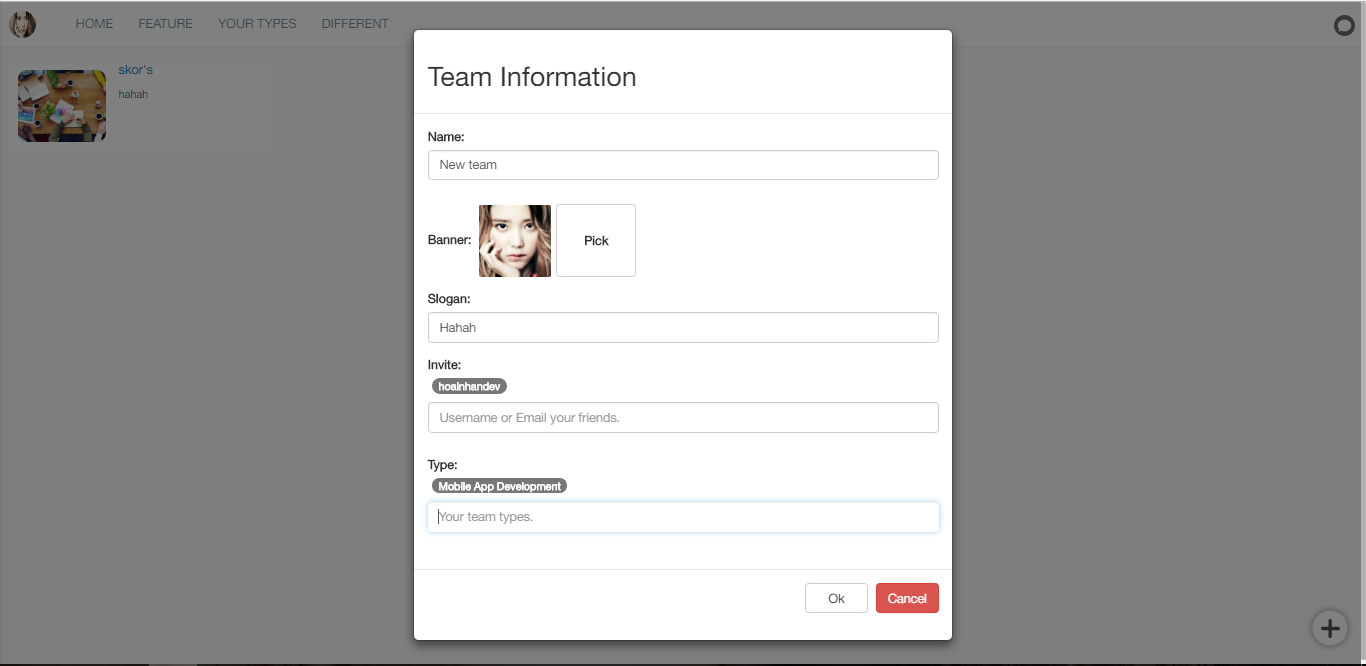
Demo . Màn hình chat 2.



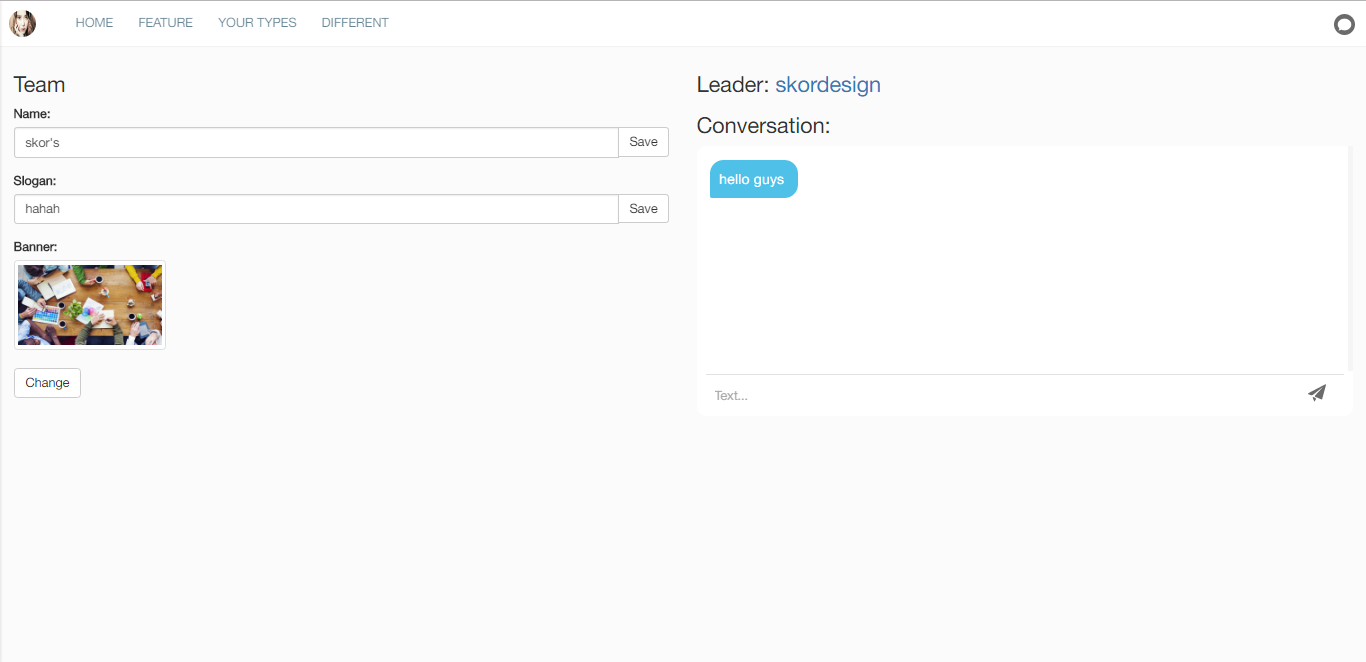
Demo . Màn hình quản lí tài khoản người dùng.



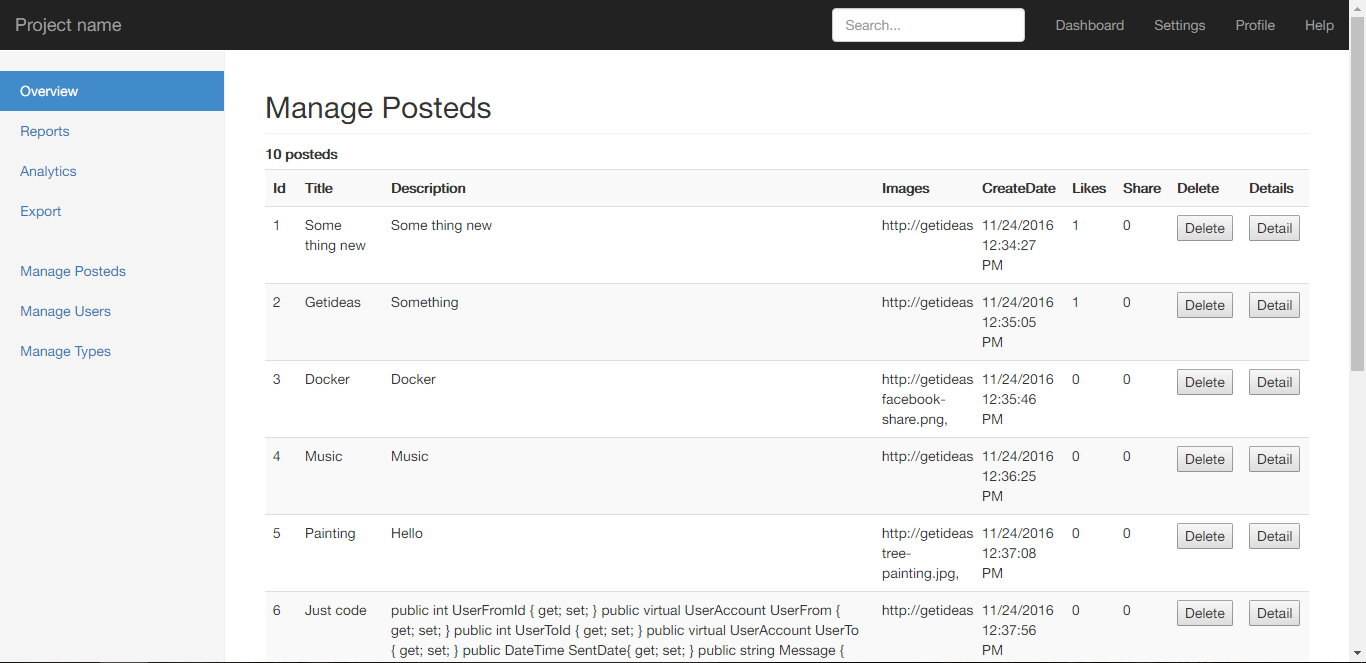
Demo . Màn hình hiển thị danh sách nhóm.



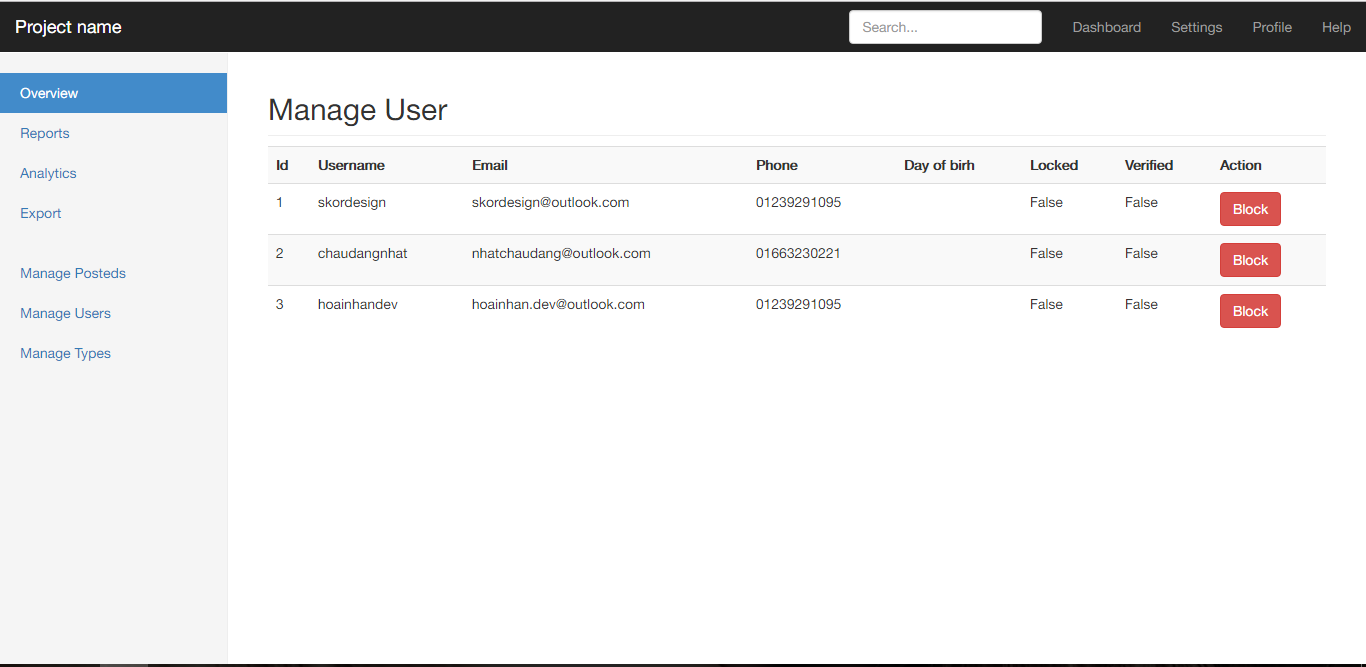
Demo . Màn hình tạo nhóm.



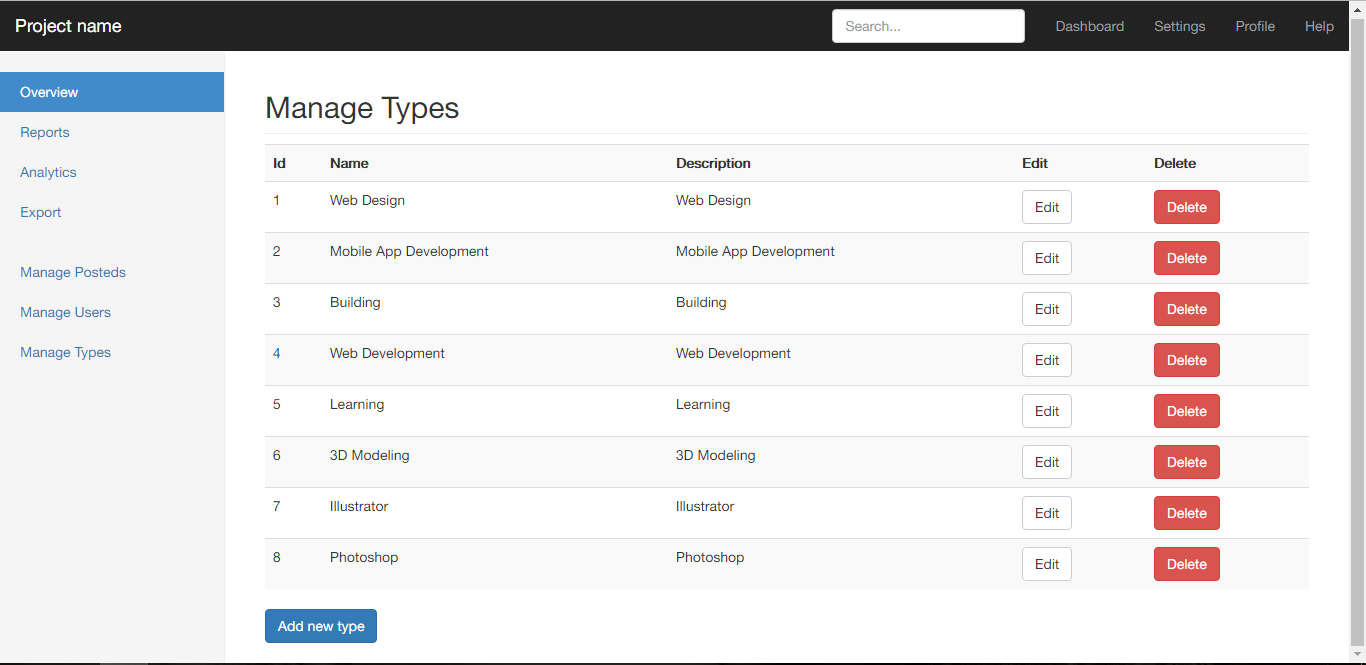
Demo . Màn hình quản lí nhóm và chat nhóm.



Demo . Màn hình quản lý bài đăng của Admin.



Demo . Màn hình quản lí tải khoản người dùng của Admin.



Demo . Màn hình quản lí danh mục của Admin.

# CHƯƠNG 4: KẾT LUẬN

Trong quá trình thực hiện đề tài có gặp một số khó khăn do tính mới mẻ của .NET Core và công nghệ kèm theo như: Entity Framework Core, Google API for .NET Core, Facebook API for .NET Core, … nhưng nhóm đã cố gắng khắc phục và áp dụng thành công. Ngoải ra, nhóm còn áp dụng framework javascript đang khá thịnh hành là AngularJS vào web nên tốc độ xử lí của trang web được cải thiện đáng kể.

Nhìn chung thì ASP.NET Core và MVC 6 với cấu trúc project được rút gọn khá là dễ hiểu so với phiên bản trước, sự xuất hiện của Tag Helpers và View Component khiến cho việc phát triển website dễ dang hơn với cú pháp ngắn gọn, dễ hiểu hơn so với Html Helpers của phiên bản trước.

# CHƯƠNG 5: HƯỚNG PHÁT TRIỂN

Trong quá trình phát triển website, tuy hoàn thành được 90% yêu cầu đề ra nhưng một số ý tưởng lại xuất hiện trong lúc làm. Sau đây là những tính năng cũng như công nghệ mà nhóm dự tính sẽ áp dụng vào website của mình trong thời gian tới.

* Data Mining: Ứng dụng Data Mining vào hệ thống để nâng cao trải nghiệm người dùng, các gợi ý cho người dùng sẽ chính xác hơn, thiết thực hơn.
* PWA: Progressive Web App là một giải pháp đa nền tảng mới của các nhà phát triển trình duyệt, nhóm đã có một số dự án làm về vấn đề này, việc áp dụng vào getIdeas chỉ là vấn đề thời gian. PWA sẽ thích hợp với tất cả các thiết bị, nền tảng khác nhau. Đồng thời dựa vào service worker, hệ thống có thể gửi thông báo đển người dùng cụ thể đồng thời website sẽ có cách hoạt động riêng khi không có internet.
* Đi sâu hơn vào chức năng làm việc nhóm: Chức năng hiện tại chỉ là cơ bản, nhóm sẽ phát triển thêm một số chức năng khác như quản lí dự án, chia task cho các thành viên trong nhóm, quản lí task, ...
* Angular 2: Phiên bản AngularJS hiện tại hệ thống đang sử dụng là phiên bản 1. Nhóm dự tính thì thay thể Angular 1 bằng Angular 2 ở thời gian tới, như thế sẽ hạn chế số lượng công việc xử lí ở server hơn, tăng tốc website lên đáng kể.
* Docker: là một công cụ đang khá thịnh hành, nhóm đang tìm hiểu và triển khai website lên Docker, như vậy sẽ tiết kiệm chi phí về sau.

# TÀI LIỆU THAM KHẢO

[1] ASP.NET Core - <https://www.asp.net/core>

<https://docs.microsoft.com/en-us/dotnet/articles/core/deploying/index>

[2] SignalR for .NET Core - <https://chsakell.com/2016/10/10/real-time-applications-using-asp-net-core-signalr-angular/>

[3] Bootstrap - <http://getbootstrap.com/components/>

<http://getbootstrap.com/css/>

[4] AngularJS - <https://docs.angularjs.org/tutorial>

[5] jQuery - <http://www.w3schools.com/jquery/>