TRƯỜNG ĐẠI HỌC GIAO THÔNG VÂN TẢI

KHOA CÔNG NGHỆ THÔNG TIN

---------------o0o---------------



**BÁO CÁO**

**THỰC TẬP CHUYÊN MÔN**

Đề tài: Nguyên cứu công nghệ ReactJS & NodeJS

Demo website Dev-Connector

Giảng viên hướng dẫn. **:** **ThS. Lương Thái Lê**

Sinh viên thực hiện **:** **Lê Trọng Khoa**

Lớp **:** **CNTT4\_K58**

Khoa **:** **Công Nghệ Thông Tin**

**NHẬN XÉT CỦA GIÁO VIÊN**

………………………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………..

Hà Nội, ngày tháng năm 2020

MỤC LỤC

DANH MỤC HÌNH VẼ............................................................................................3

CHƯƠNG 1. MỞ ĐẦU.............................................................................................5

1. Lý do chọn đề tài: ...........................................................................................5
2. Ưu khuyết điểm...............................................................................................5
   1. 2.1.  Ưu điểm ....................................................................................... 5
   2. 2.2.  Khuyết điểm ................................................................................ 5
3. Ý nghĩa lý luận và ý nghĩa thực tiễn của đề tài :.............................................5

CHƯƠNG 2:TỔNG QUAN VỀ CÁC FRAMEWORK VÀ CÁC THƯ VIỆN CÓ TRONG PROJECT DEMO......................................................................................6

1. ReactJS: .........................................................................................................6
2. Redux…………..............................................................................................9
3. NodeJS..........................................................................................................12
4. Cơ sở dữ liệu mongoDb................................................................................13

CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ CHƯƠNG TRÌNH...................................16

KẾT LUẬN ............................................................................................................23

TÀI LIỆU THAM KHẢO ......................................................................................24

**Lời mở đầu**

* **JavaScript** là ngôn ngữ lập trình phổ biến nhất trên thế giới trong suốt 20 năm qua. Nó cũng là một trong số 3 ngôn ngữ chính của lập trình web: HTML, CSS, javascript.
* Với sự phát triển công nghệ yêu cầu về sản phẩm công nghệ ngày càng nhiều, vì thế các công nghệ, framework, thư viện ra đời nhiều hơn để giúp lập trình viên làm việc hiểu quả hơn, hơn thế nữa nó còn tối ưu hoá tốc độ chạy render load trang web điều mà người dung rất quan tâm.
* Trong lần nghiên cứu bài tập lần này em đã chọn làm frameword reactJs phía client và nodeJS làm server vì nó phát triển kĩ năng cũng như mong muốn sau này của em sẽ trở thành full stack javascript.
* Trong quá trình làm đề tài không thể tránh khỏi sai sót. Rất mong thầy và các bạn góp ý để bài tập em được hoàn thiện hợn.
* Một lần nữa xin chân thành cảm ơn.

# CHƯƠNG 1. MỞ ĐẦU

1. **Lý do chọn đề tài:**

Sau tìm hiểu và trải nghiệm với kinh nghiệm gần 1 năm làm việc và đóng gói ý kiến của GVHD dẫn em đã lựa chọn “**NGHIÊN CỨU FRAMEWORK REACTJS & NODEJS DEMO DEV-CONNECTOR**” để làm project với lý do:

* Tìm hiểu công nghệ mới hiện nay và để xây dựng và tích luỹ kiến thức phục vụ cho công việc và mong muốn làm việc sau này.
* Project với sự lựa chọn client side ReactJS, server side NodeJS, cơ sở dữ liệu Mongodb, và một số thư viện khác

1. **Ưu khuyết điểm**

### **Ưu điểm**

* Đề tài giúp phát triển skill cá nhân cũng như khả năng tự tìm hiểu.
* Website là một single page application có tốc độ truyền tải rất nhanh.
* Xây dựng giao diện khách hàng bắt mắt có tình năng hay như cộng đồng developer thu nhỏ, có like comment post trạng thái, bình luận, xem hồ sơ lập trình viên, các repository github
  1. **Khuyết điểm**

- Do thời gian có hạn vừa phải tìm hiểu lắm chắc kiến thức để có thể xây dựng web demo nên chức năng còn chưa được tối ưu, chưa realtime chat comment like

1. **Ý tưởng lý luận và ý nghĩa thực tiễn của đề tài :**

* Đề tài đóng vai trò quan trọng trong việc nâng cấp kiến thức và tiếp cận dự án thực tế
* Nâng cao khả năng độc lập làm việc – cũng là một sản phẩm điền vào CV công việc sau này.

# CHƯƠNG 2: TỔNG QUAN VỀ CÁC FRAMWORK VÀ THƯ VIỆN NGHIÊN CÓ TRONG PROJECT

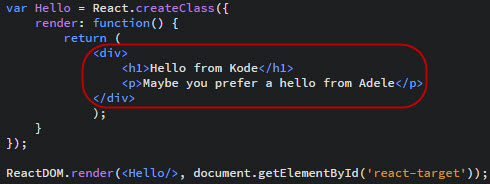
1. **ReactJS**
   1. **Tổng quan:**

Trong thời đại công nghệ hóa hiện nay dưới sự phát triển vượt bậc của ngành công nghệ điện tử kéo theo những ngành nghề liên quan cũng phát triển trong đó có cả việc sử dụng những thư viện JavaScript cho các trang website. Vì thế ReactJS đã ra đời và trở nên phổ biến bởi tính năng linh hoạt đơn giản mà nó đem lại. Theo thống kê con số sử dụng ReactJS tính tới hiện nay là 1300 developer và hơn 94000 trang web đang sử dụng nó. Vậy nay em đã tìm hiểu về ReactJS.

* 1. **ReactJs là gì?**
* ReactJS được hiểu là một thư viện trong đó có chứa nhiều JavaScript mã nguồn mở và cha đẻ của ReactJS đó chính là một ông lớn với cái tên ai cũng biết đó chính là Facebook. Mục đích của việc tạo ra ReactJS là để tạo ra những ứng dụng website hấp dẫn với tốc độ nhanh và hiệu quả cao với những mã hóa tối thiểu. Và mục đích chủ chốt của ReactJS đó chính là mỗi website khi đã sử dụng ReactJS thì phải chạy thật mượt thật nhanh và có khả năng mở rộng cao và đơn giản thực hiện.
* Nhìn chung tất cả những tính năng hay sức mạnh của ReactJS thường xuất phát từ việc tập trung vào các phần riêng lẻ chính vì điểm này nên khi làm việc trên web thay vì nó sẽ làm việc trên toàn bộ ứng dụng của website thì ReactJS cho phép developer có chức năng phá vỡ giao diện của người dùng từ một cách phức tạp và biến nó trở thành các phần đơn giản hơn nhiều lần có nghĩa là render dữ liệu không chỉ được thực hiện ở vị trí sever mà còn có thể thực hiện ở vị trí Client khi sử dụng ReactJS.
  1. **Đặc trưng của reactJs:**

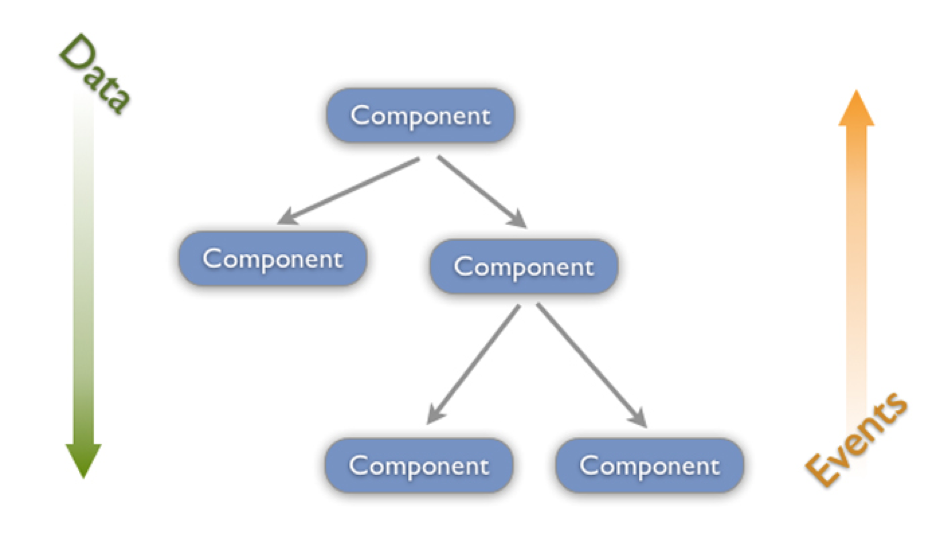
Reactjs là có rất nhiều đặc trưng và em sẽ giơi thiệu đặc trung nổi bật:

**JSX**



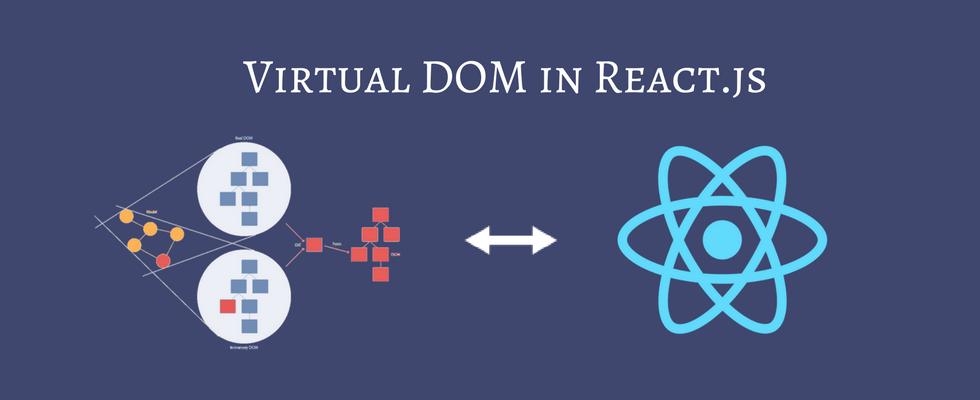
Trong React, thay vì thường xuyên sử dụng JavaScript để thiết kế bố cục trang web thì sẽ dùng JSX. JSX được đánh giá là sử dụng đơn giản hơn JavaScript và cho phép trích dẫn HTML cũng như việc sử dụng các cú pháp thẻ HTML để render các subcomponent. JSX tối ưu hóa code khi biên soạn, vì vậy nó chạy nhanh hơn so với code JavaScript tương đương.

**Single-way data follow (Dữ liệu 1 chiều)**



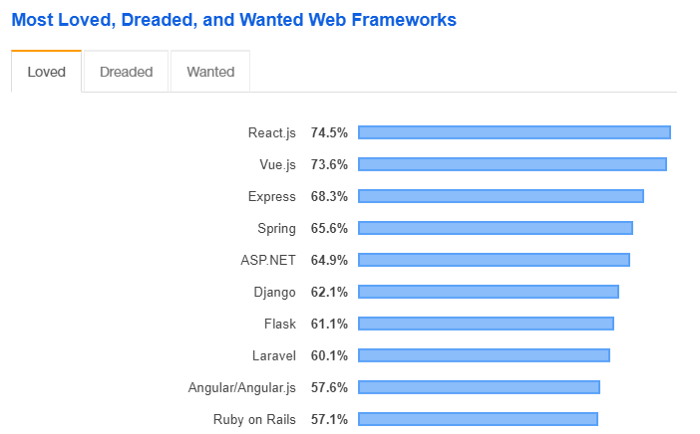
ReactJS không có những module chuyên dụng để xử lý data, vì vậy ReactJS chia nhỏ view thành các component nhỏ có mỗi quan hệ chặt chẽ với nhau. Tại sao phải quan tâm tới cấu trúc và mối quan hệ giữa các component trong ReactJS? Câu trả lời chính là luồng truyền dữ liệu trong ReactJS: Luồng dữ liệu một chiều từ cha xuống con. Việc ReactJS sử dụng one-way data flow có thể gây ra một chút khó khăn cho những người muốn tìm hiểu và ứng dụng vào trong các dự án. Tuy nhiên, cơ chế này sẽ phát huy được ưu điểm của mình khi cấu trúc cũng như chức năng của view trở nên phức tạp thì ReactJS sẽ phát huy được vai trò của mình.

**Virtual Dom**



Những Framework sử dụng Virtual-DOM như ReactJS khi Virtual-DOM thay đổi, chúng ta không cần thao tác trực tiếp với DOM trên View mà vẫn phản ánh được sự thay đổi đó. Do Virtual-DOM vừa đóng vai trò là Model, vừa đóng vai trò là View nên mọi sự thay đổi trên Model đã kéo theo sự thay đổi trên View và ngược lại. Có nghĩa là mặc dù chúng ta không tác động trực tiếp vào các phần tử DOM ở View nhưng vẫn thực hiện được cơ chế Data-binding. Điều này làm cho tốc độ ứng dụng tăng lên đáng kể – môt lợi thế không thể tuyệt vời hơn khi sử dụng Virtula-DOM.

**Tại sao ReactJs là lựa chọn hàng đầu hiện nayc của doanh nghiệp?**



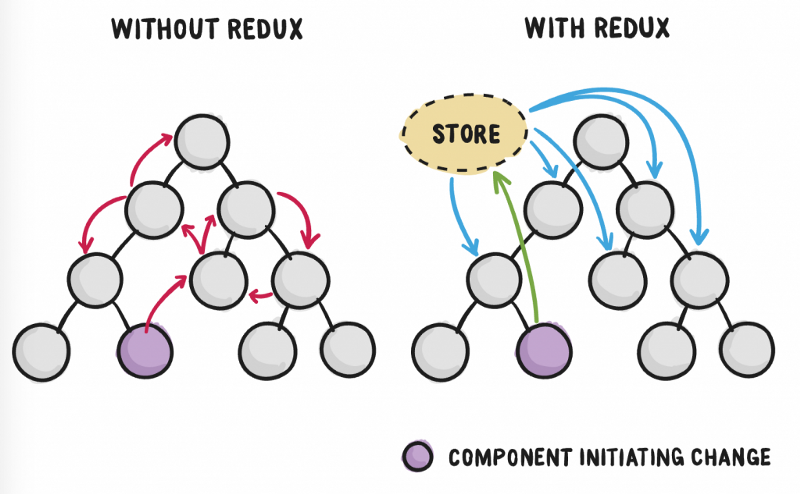
Biểu đồ năm 2019 của stackoverflow

Trong lĩnh vực phát triển công nghệ, các chủ doanh nghiệp và developer luôn tìm kiếm những phương pháp tốt nhất để giúp doanh nghiệp của họ có những lợi thế cạnh tranh tốt hơn. Và một trong những công nghệ tốt nhất có thể giúp những doanh nghiệp vượt mặt đối thủ trong việc tạo ra những ứng dụng web chính là ReactJS.

ReactJS cho phép các doanh nghiệp tạo ra những ứng dụng web với UI tốt hơn để nâng cao trải nghiệm người dùng. Đây cũng chính là công nghệ mà các doanh nghiệp cần để có được lượng tương tác của người dùng, tỉ lệ click cũng như chuyển đổi cao hơn. Hơn thế, các doanh nghiệp sử dụng ReactJS được đảm bảo có giao diện tốt hơn so với những doanh nghiệp sử dụng các framework khác bởi ReactJS giúp ngăn chặn việc cập nhật của DOM giúp ứng dụng nhanh hơn và truyền tải tốt hơn UX.

1. **Redux**
   1. **Redux là gì?**

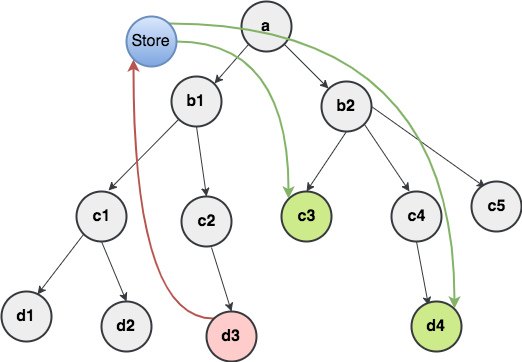
**Redux** là một predictable state management tool cho các ứng dụng Javascript. Nó giúp bạn viết các ứng dụng hoạt động một cách nhất quán, chạy trong các môi trường khác nhau (client, server, and native) và dễ dàng để test. **Redux** ra đời lấy cảm hứng từ tư tưởng của ngôn ngữ **Elm** và kiến trúc **Flux**của Facebook. Do vậy Redux thường dùng kết hợp với React.



* 1. **Nguyên lý hoạt động**

Cái cách mà Redux hoạt động là khá đơn giản. Nó có 1 store lưu trữ toàn bộ state của app. Mỗi component có thể access trực tiếp đến state được lưu trữ thay vì phải send drop down props từ component này đến component khác.

Có 3 thành phần của Redux: Actions, Store, Reducers.



* Actions:

Actions đơn giản là các events. Chúng là cách mà chúng ta send data từ app đến Redux store. Những data này có thể là từ sự tương tác của user vs app, API calls hoặc cũng có thể là từ form submission.

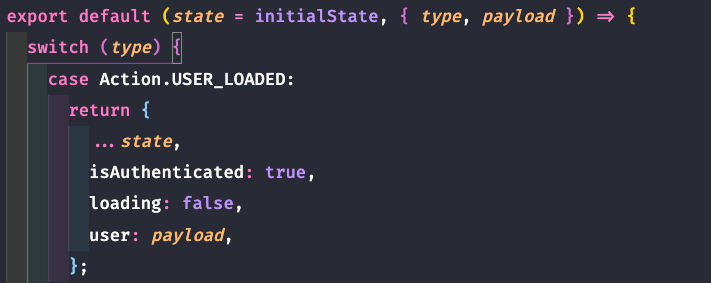
Actions được gửi bằng cách sử dụng store.dispatch() method, chúng phải có một type property để biểu lộ loại action để thực hiện. Chúng cũng phải có một payload chứa thông tin. Actions được tạo thông qua một action creator. Ví dụ



* Reduces:

Reducers là các function nguyên thủy chúng lấy state hiện tại của app, thực hiện một action và trả về một state mới. Những states này được lưu như những objects và chúng định rõ cách state của một ứng dụng thay đổi trong việc phản hồi một action được gửi đến store.

Đây là một ví dụ về cách mà Reducers hoạt động trong Redux:



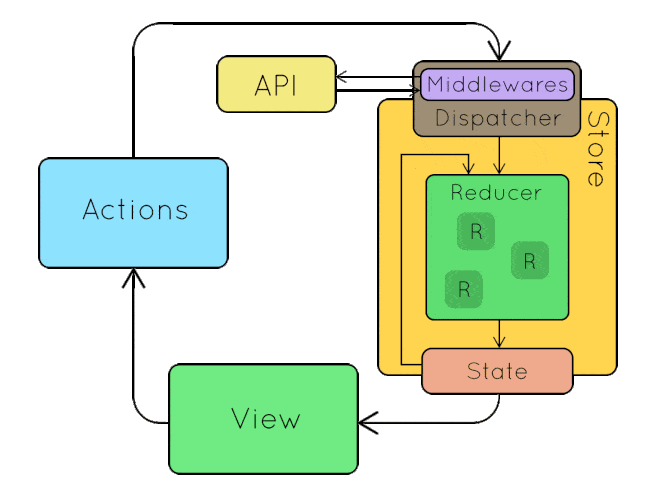
* Store

Store lưu trạng thái ứng dụng và nó là duy nhất trong bất kỳ một ứng dụng Redux nào. Bạn có thể access các state được lưu, update state, và đăng ký or hủy đăng ký các listeners thông qua helper methods.

Tạo một store cho một login app:

Const store = createStore(loginConponent);(LoginComponent);

Nguyên lý vận hành:



1. **NodeJS**
   1. **Khái niệm về NodeJS**

* Nodejs là một **nền tảng** (Platform) phát triển độc lập được xây dựng ở trên Javascript Runtime của Chrome mà chúng ta có thể xây dựng được các ứng dụng mạng một cách nhanh chóng và dễ dàng mở rộng.
* Nodejs được xây dựng và phát triển từ năm 2009, bảo trợ bởi công ty Joyent, trụ sở tại California, Hoa Kỳ. Dù sao thì chúng ta cũng nên biết qua một chút chút lịch sử của thứ mà chúng ta đang học một chút chứ nhỉ?
* Phần Core bên dưới của Nodejs được viết hầu hết bằng C++ nên cho tốc độ xử lý và hiệu năng khá cao.
* Nodejs tạo ra được các ứng dụng có tốc độ xử lý nhanh, realtime thời gian thực.
* Nodejs áp dụng cho các sản phẩm có lượng truy cập lớn, cần mở rộng nhanh, cần đổi mới công nghệ, hoặc tạo ra các dự án Startup nhanh nhất có thể.
  1. **Lý do sử dụng NodeJS**
* Các ứng dụng Nodejs được viết bằng **javascript**, ngôn ngữ này là một ngôn ngữ khá thông dụng. Theo tác giả của ngôn ngữ Javascript, Ryan Dahl: *“Javascript có những đặc tính mà làm cho nó rất khác biệt so với các ngôn ngữ lập trình động còn lại, cụ thể là nó không có khái niệm về đa luồng, tất cả là đơn luồng và hướng sự kiện.”*
* Nodejs chạy đa nền tảng phía Server, sử dụng kiến trúc hướng sự kiện Event-driven, cơ chế non-blocking I/O làm cho nó nhẹ và hiệu quả.
* Có thể chạy ứng dụng Nodejs ở bất kỳ đâu trên máy Mac – Window – Linux, hơn nữa cộng đồng Nodejs rất lớn và hoàn toàn miễn phí. Các bạn có thể thấy cộng đồng Nodejs lớn như thế nào tại đây, các package đều hoàn toàn free: <https://www.npmjs.com/>
* Các ứng dụng NodeJS đáp ứng tốt thời gian thực và chạy đa nền tảng, đa thiết bị.

1. **Cơ sở dữ liệu MongoDB**
   1. **NoSql**

Đầu tiên về cơ sở dữ liệu phi quan hệ chúng ta cùng tìm hiểu về NoSql, vậy NoSql là gì?.

* NoSQL là 1 dạng CSDL mã nguồn mở và được viết tắt bởi: None-Relational SQL hay có nơi thường gọi là Not-Only SQL.
* NoSQL được phát triển trên Javascript Framework với kiểu dữ liệu là JSON và dạng dữ liệu theo kiểu key và value.
* NoSQL ra đời như là 1 mảnh vá cho những khuyết điểm và thiếu xót cũng như hạn chế của mô hình dữ liệu quan hệ RDBMS (Relational Database Management System - Hệ quản trị cơ sở dữ liệu quan hệ) về tốc độ, tính năng, khả năng mở rộng,...
* Với NoSQL bạn có thể mở rộng dữ liệu mà không lo tới những việc như tạo khóa ngoại, khóa chính, kiểm tra ràng buộc .v.v ...
* NoSQL bỏ qua tính toàn vẹn của dữ liệu và transaction để đổi lấy hiệu suất nhanh và khả năng mở rộng.
* NoSQL được sử dụng ở rất nhiều công ty, tập đoàn lớn, ví dụ như FaceBook sử dụng Cassandra do FaceBook phát triển, Google phát triển và sử dụng BigTable,...
  1. **MongoDb là gì?**
* MongoDB là một hệ quản trị cơ sở dữ liệu mã nguồn mở, là CSDL thuộc NoSql và được hàng triệu người sử dụng.
* MongoDB là một database hướng tài liệu (document), các dữ liệu được lưu trữ trong document kiểu JSON thay vì dạng bảng như CSDL quan hệ nên truy vấn sẽ rất nhanh.
* Với CSDL quan hệ chúng ta có khái niệm bảng, các cơ sở dữ liệu quan hệ (như MySQL hay SQL Server...) sử dụng các bảng để lưu dữ liệu thì với MongoDB chúng ta sẽ dùng khái niệm là **collection** thay vì bảng
* So với RDBMS thì trong MongoDB **collection** ứng với **table**, còn **document** sẽ ứng với **row** , MongoDB sẽ dùng các document thay cho row trong RDBMS.
* Các collection trong MongoDB được cấu trúc rất linh hoạt, cho phép các dữ liệu lưu trữ không cần tuân theo một cấu trúc nhất định.
* Thông tin liên quan được lưu trữ cùng nhau để truy cập truy vấn nhanh thông qua ngôn ngữ truy vấn MongoDB
  1. **Các kiểu dữ liệu trong mongoDb**

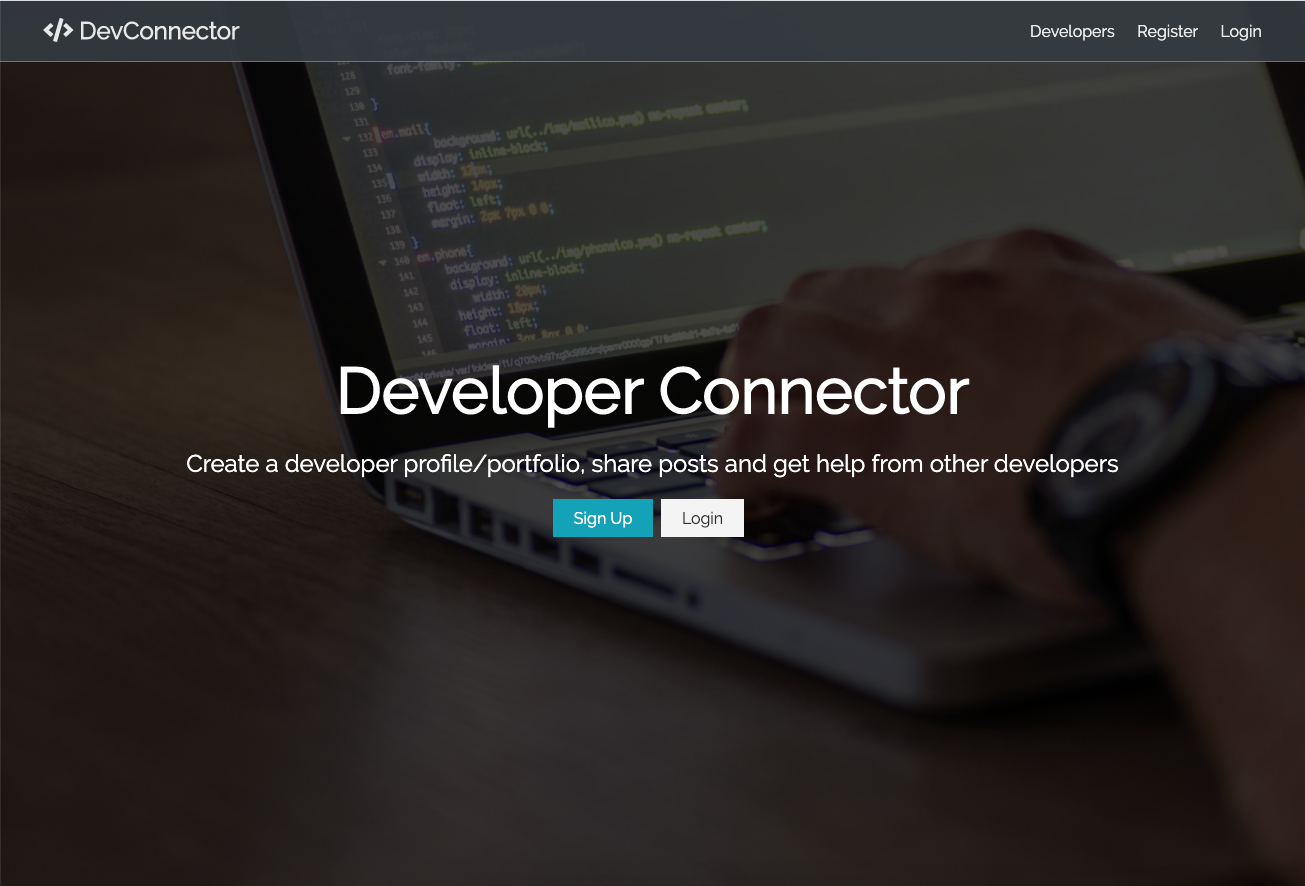
MongoDB hỗ trợ các kiểu dữ liệu sau:

* **Chuỗi:** Đây là kiểu dữ liệu được sử dụng phổ biến nhất để lưu giữ dữ liệu. Chuỗi trong MongoDB phải là UTF-8 hợp lệ.
* **Số nguyên:** Kiểu dữ liệu này được sử dụng để lưu một giá trị số. Số nguyên có thể là 32 bit hoặc 64 bit phụ thuộc vào Server của bạn.
* **Boolean**: Kiểu dữ liệu này được sử dụng để lưu giữ một giá trị Boolean (true/false).
* **Double**: Kiểu dữ liệu này được sử dụng để lưu các giá trị số thực dấu chấm động.
* **Min/ Max keys:** Kiểu dữ liệu này được sử dụng để so sánh một giá trị với các phần tử BSON thấp nhất và cao nhất.
* **Mảng:** Kiểu dữ liệu này được sử dụng để lưu giữ các mảng hoặc danh sách hoặc nhiều giá trị vào trong một key.
* **Timestamp:** Đánh dấu thời điểm một Document được sửa đổi hoặc được thêm vào.
* **Object:** Kiểu dữ liệu này được sử dụng cho các Document được nhúng vào.
* **Null:** Kiểu dữ liệu này được sử dụng để lưu một giá trị Null.
* **Symbol:** Kiểu dữ liệu này được sử dụng giống như một chuỗi
* **Date :** Kiểu dữ liệu này được sử dụng để lưu giữ date và time hiện tại trong định dạng UNIX time.
* **Object ID:** Kiểu dữ liệu này được sử dụng để lưu giữ ID của Document.
* **Binary data:** Kiểu dữ liệu này được sử dụng để lưu giữ dữ liệu nhị phân.
* **Code:** Kiểu dữ liệu này được sử dụng để lưu giữ JavaScrip code vào trong Document.
* **Regular expression:** Kiểu dữ liệu này được sử dụng để lưu giữ Regular Expresion.

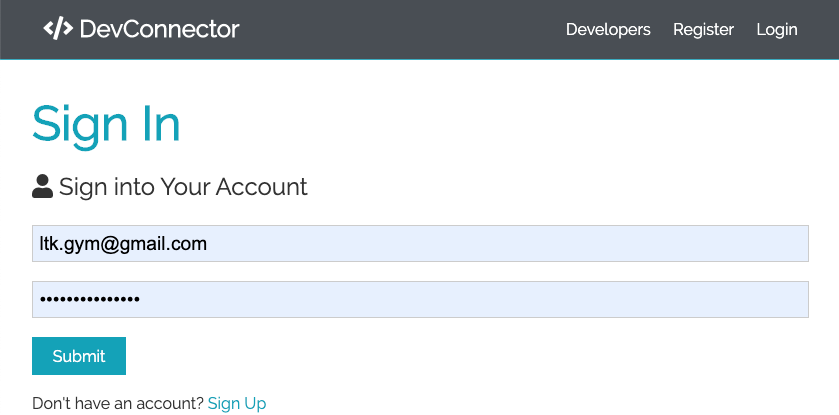
Tỉm hiểu thêm trên document

CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ CHƯƠNG TRÌNH

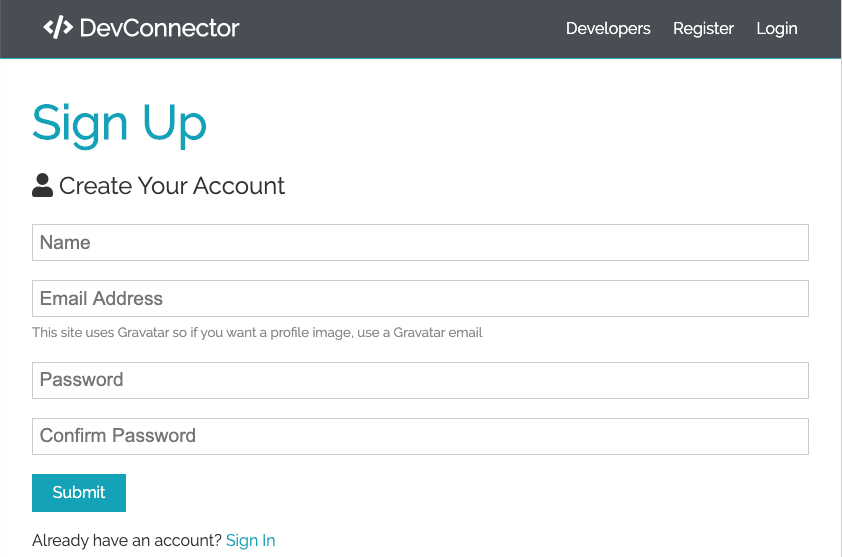
1. **Giao diện trang chủ: Giới thiệu mô tả về trang web**



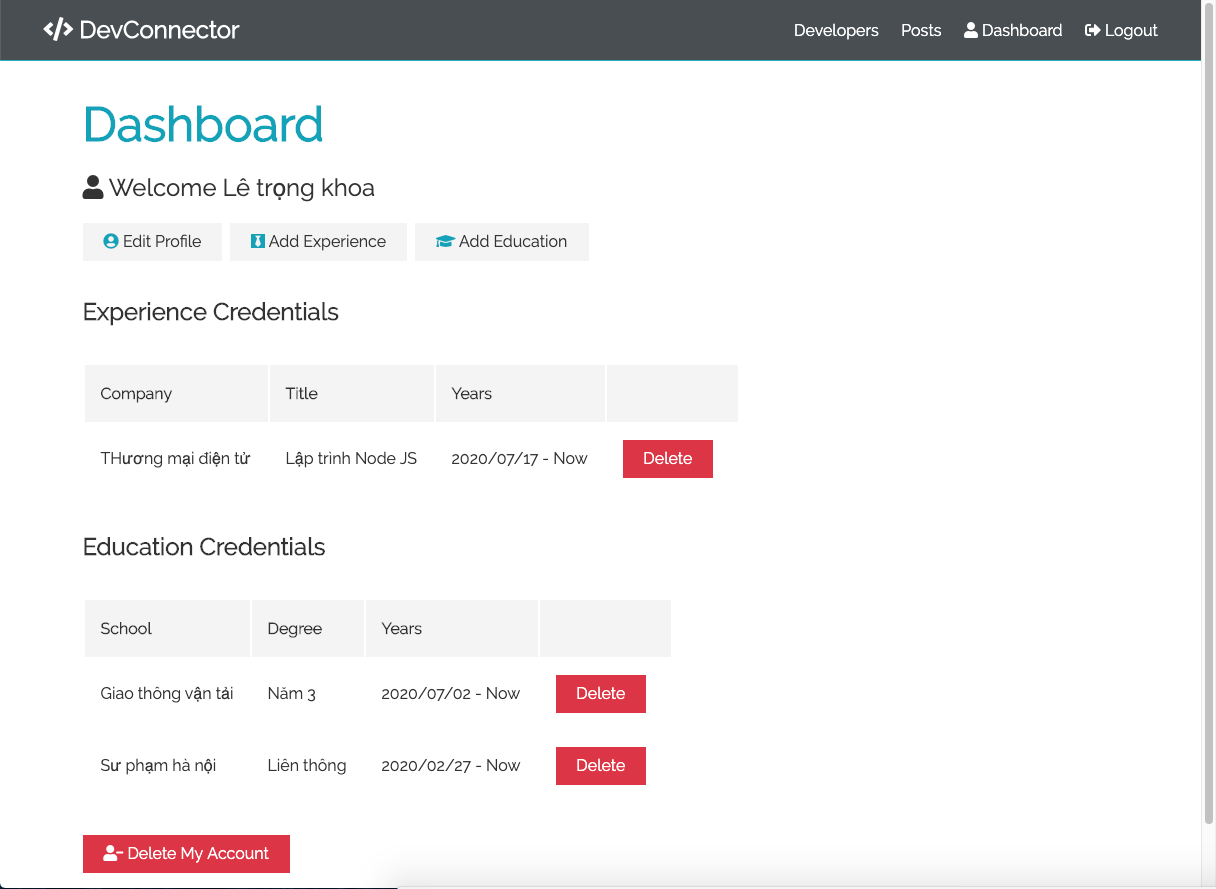
1. Giao diện đăng nhập: Xác thực tài khoản để vào trang web với các chức năng dành riêng cho người dung.

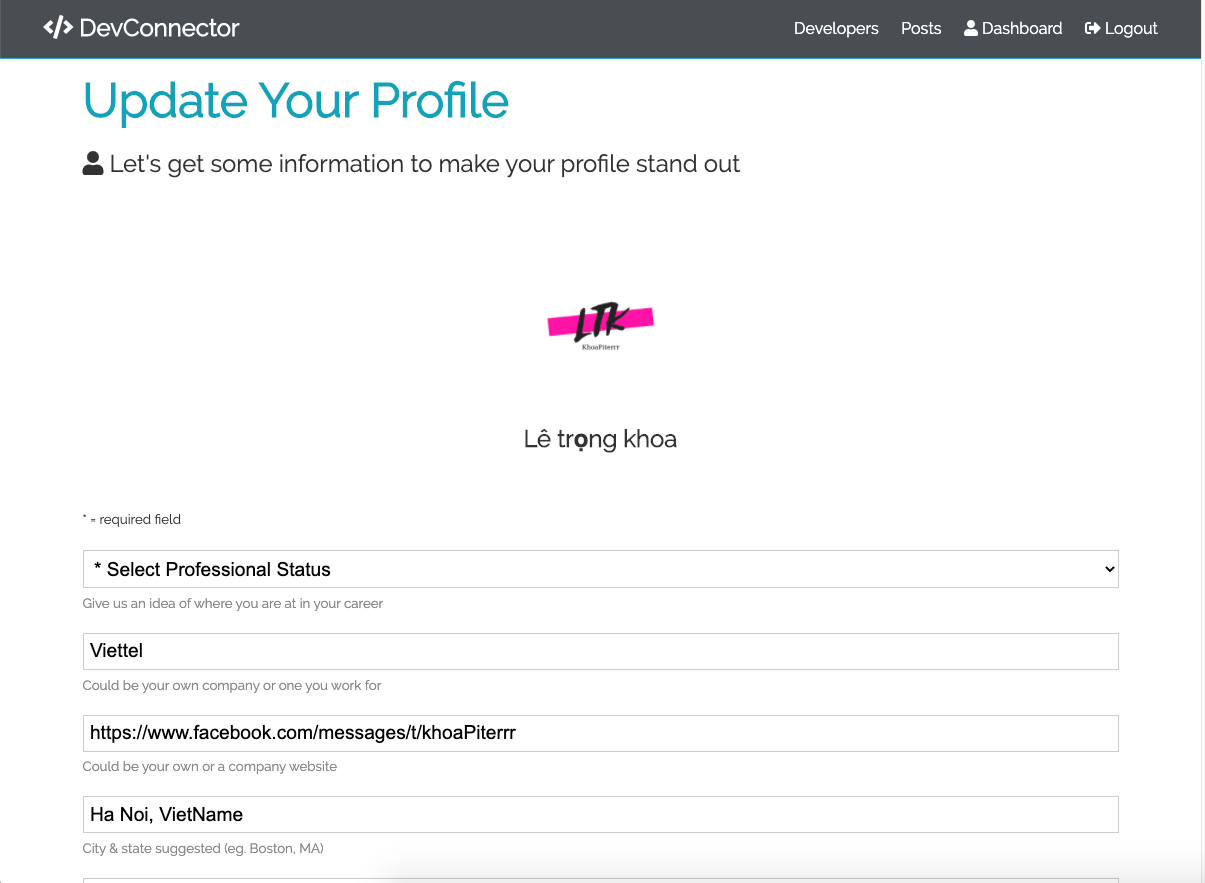


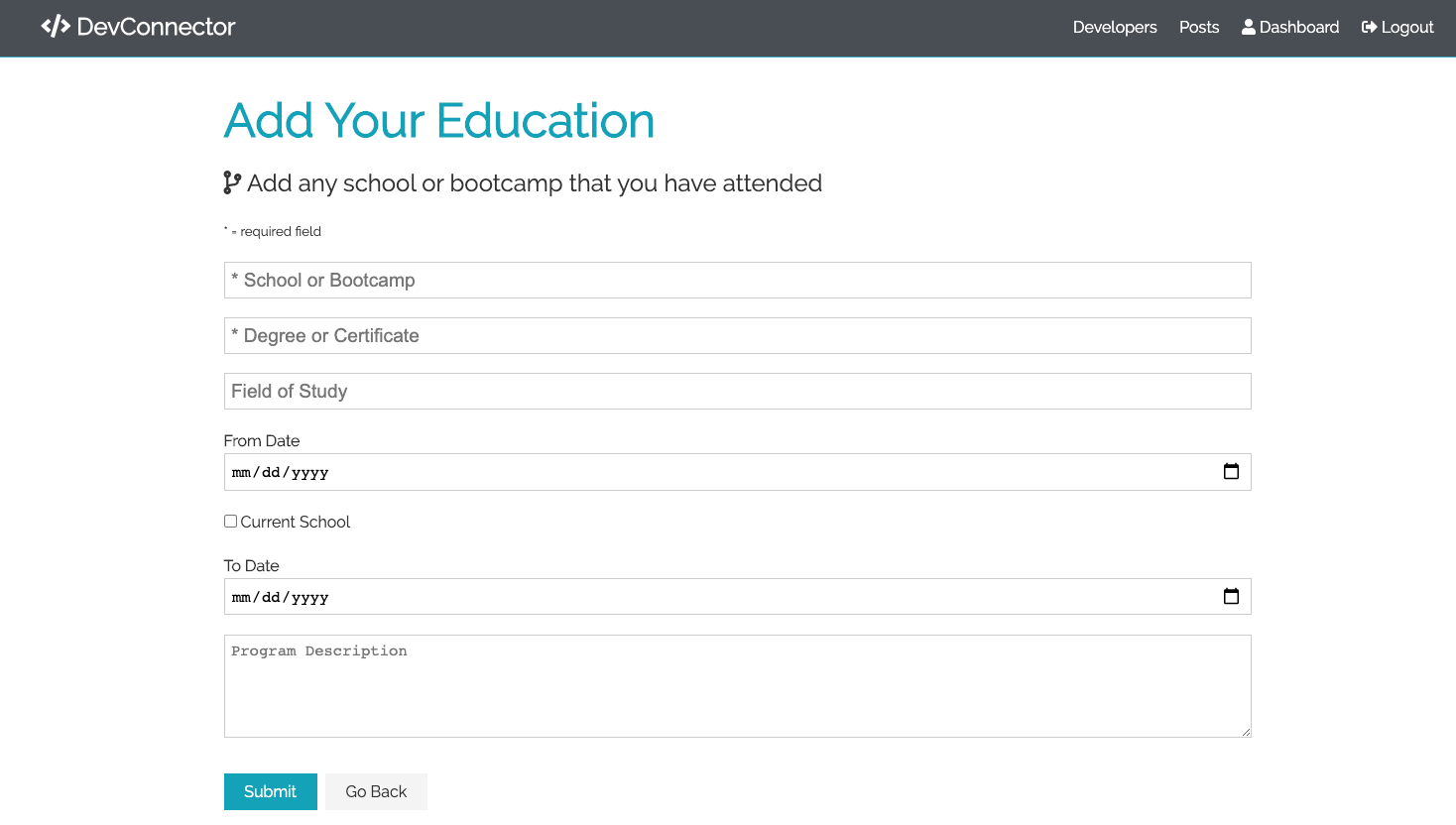
1. Giao diện đăng kí: Đăng kí tài khoản để đăng nhập web

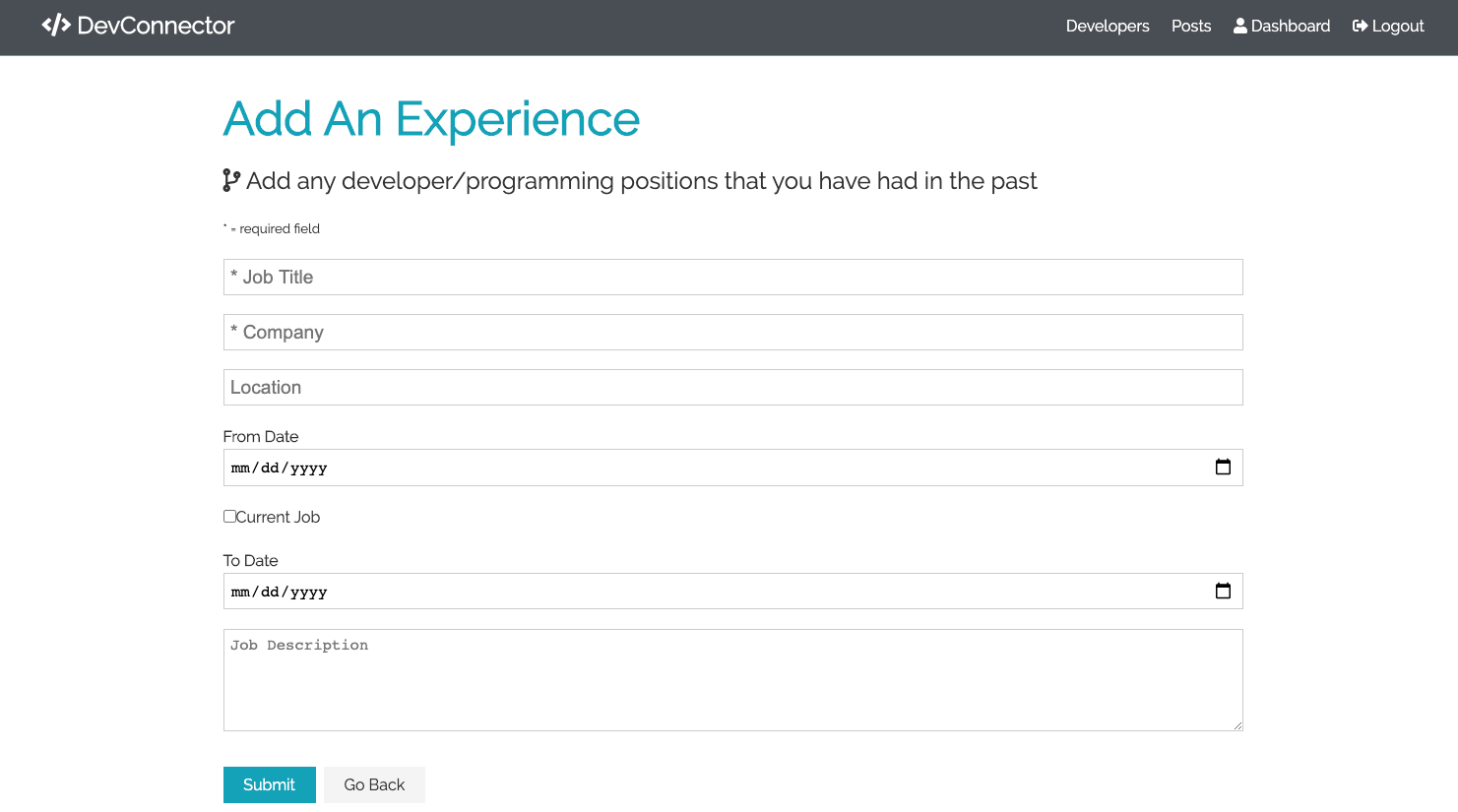


1. Dashboard: Quản lý tất cả thông tin chung của người dùng, cập nhật thông tin người dùng, thêm mới kinh nghiệm và học vấn.

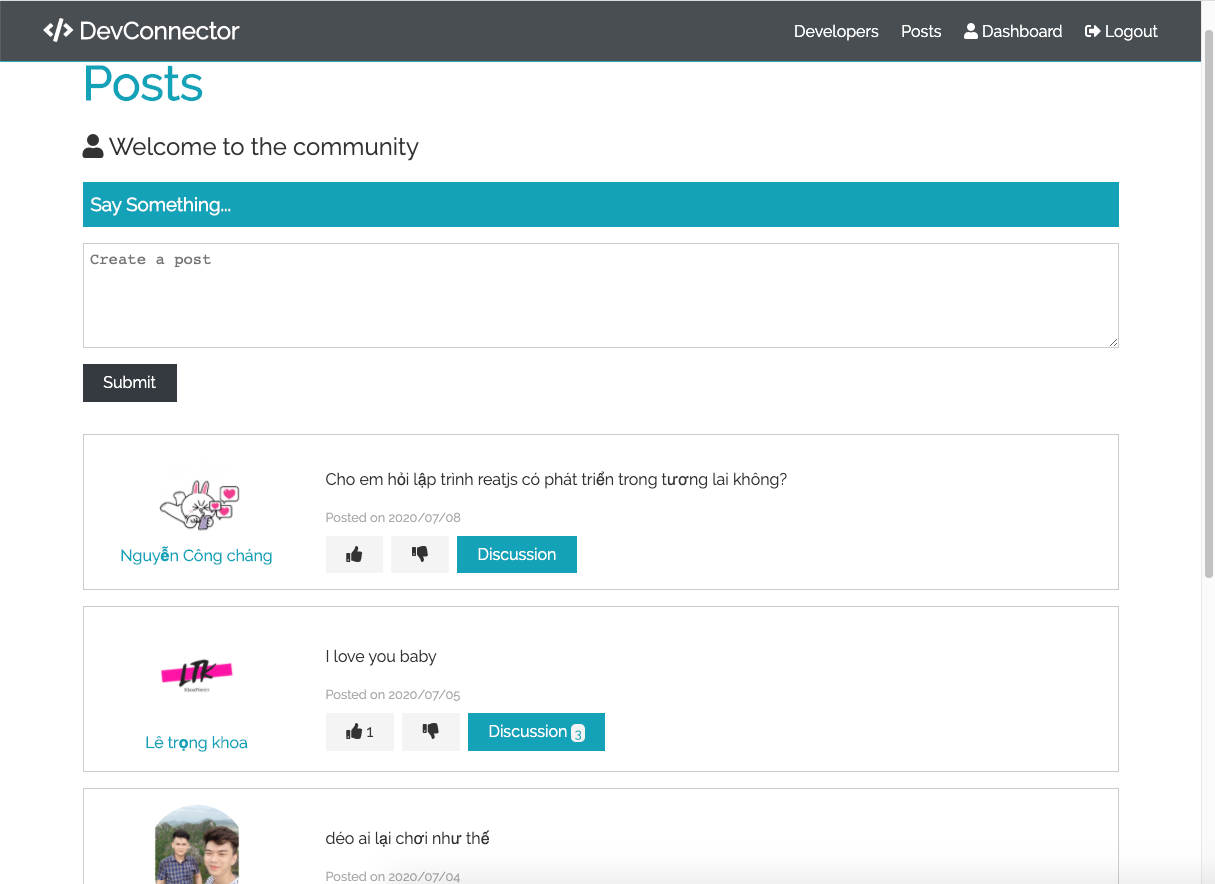


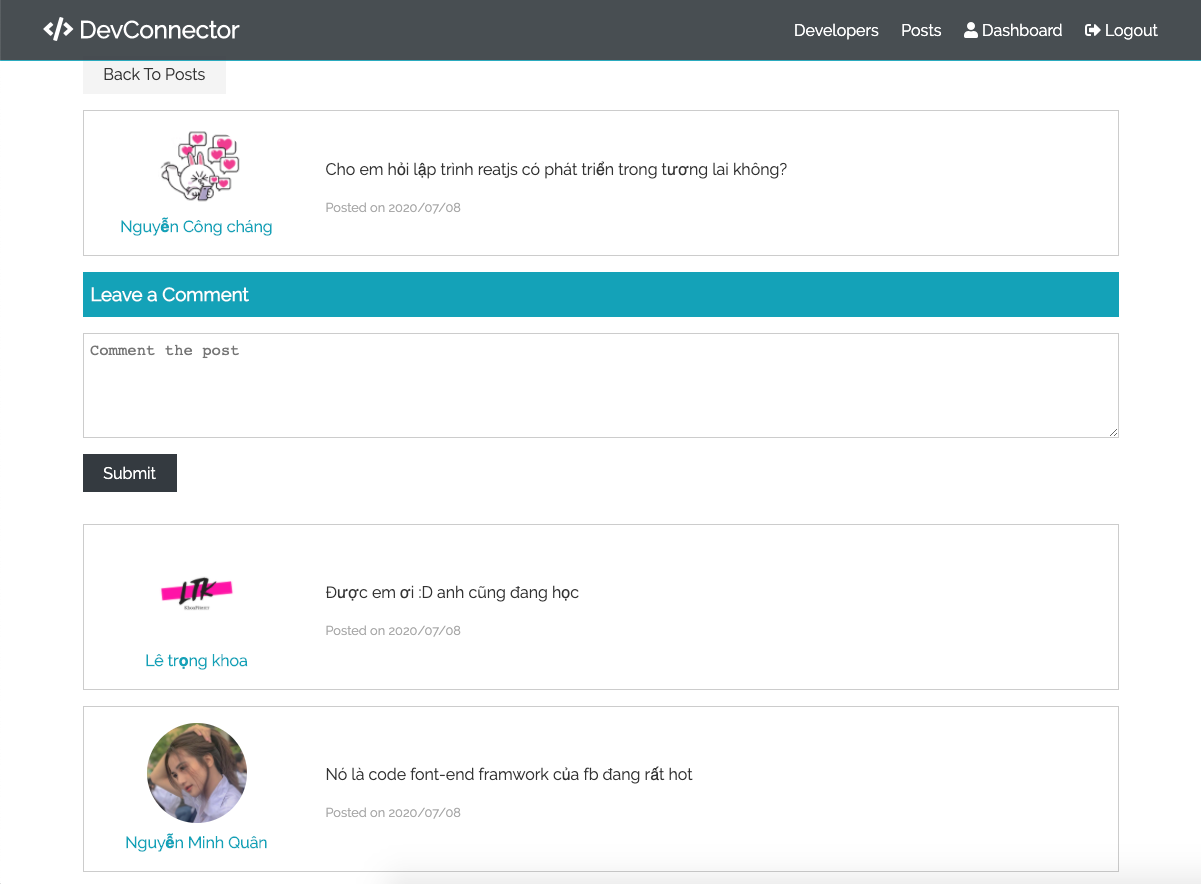




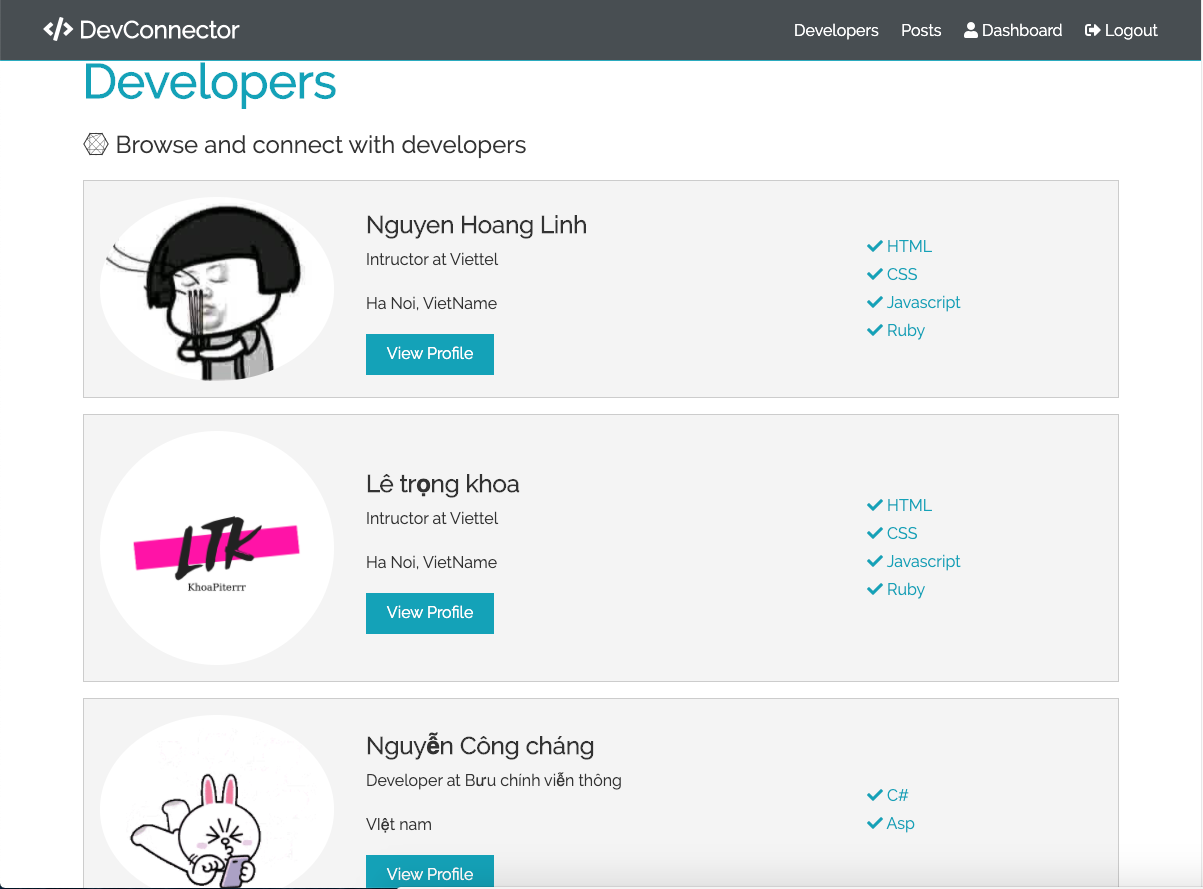


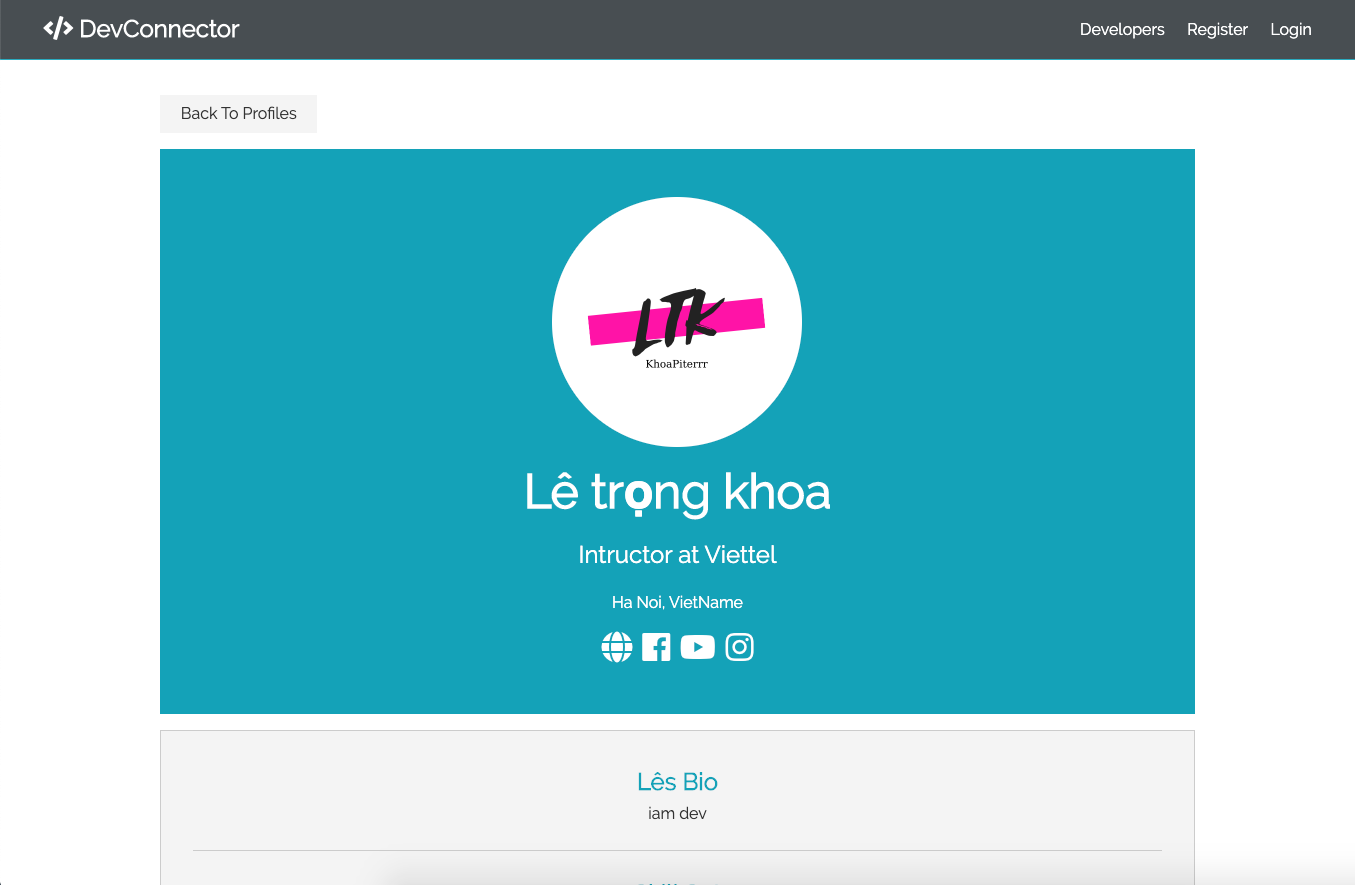
1. Giao diện đăng bài hỏi đáp, like, comment, trả lời comment cho các develop.

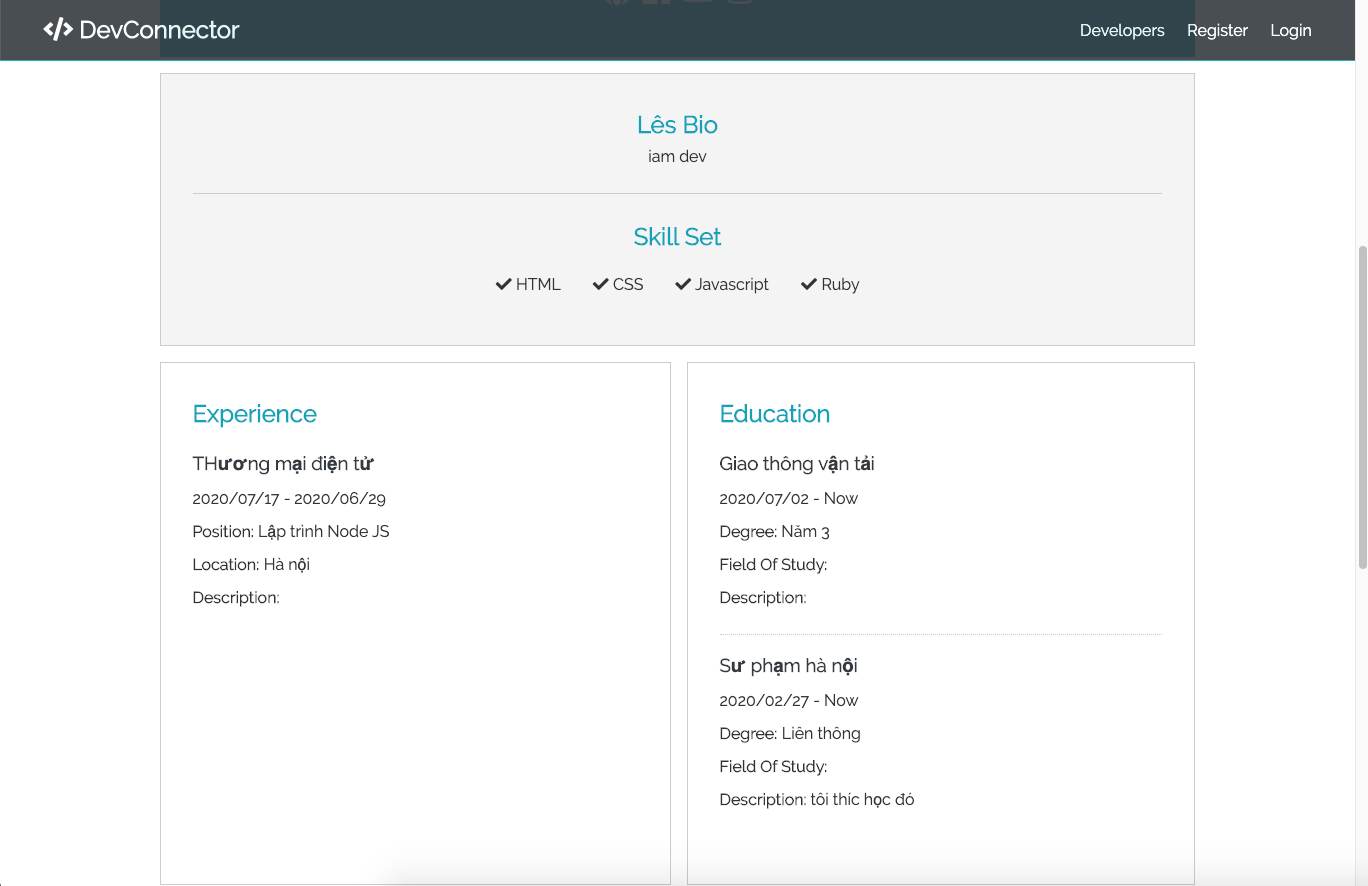


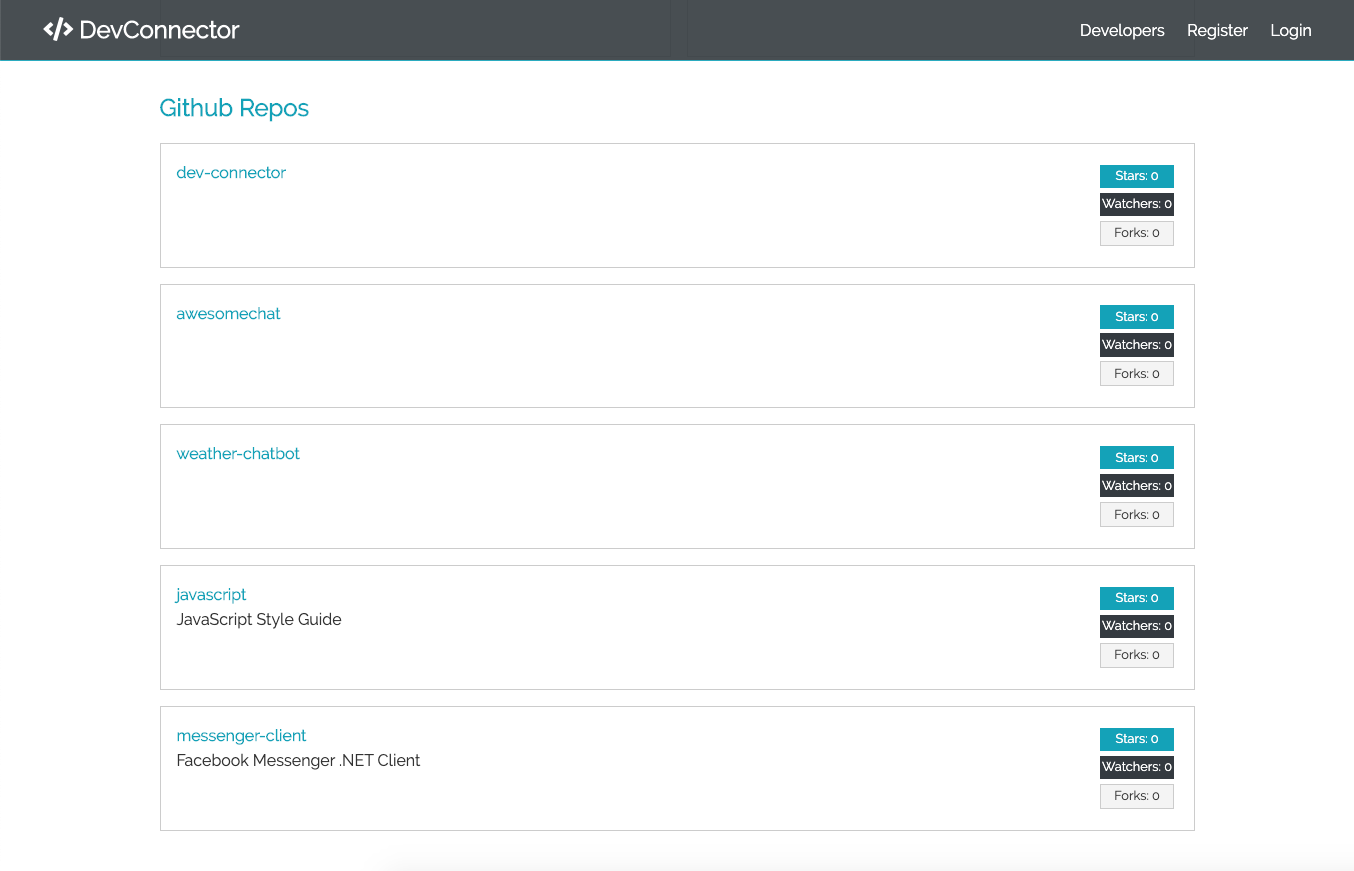


1. Giao diện xem danh sách profile của develop và xem chi tiết, có sử dụng api của github để sử lý lấy dự án đã đăng tải trên github để view









**KẾT LUẬN**

1. Kết quả đạt được của đề tài:

* -  Về cơ bản, chương trình đã hoàn thành các chức năng và yêu cầu đặt ra theo mong muốn.
* -  Giao diện chương trình khá thân thiện với người dùng.
* -  Các chức năng dễ sử dụng thân thiện như mạng xã hội bình thường.
* -  Qua đề tài này, đã thu được những kiến thức cơ bản về javascript, đặc biệt biết thêm NodeJs, ReactJs, Redux, MongoDb, cấu trúc tổ chức chương trình.
* -  Trau dồi thêm kiến thức về quy trình phát triển phần mềm, phân tích thiết kế hướng đối tượng.
* - Tự tin hơn với kiến thức javascript đã có ở mức căn bản và sẽ học thêm nâng cao để hoàn thiện bản hơn.

1. Hạn chế của đề tài :
   * -  Hệ thống mới chỉ ở quy mô nhỏ, hẹp, chưa mở rộng sự quản lý của admin.
   * -  Còn nhiều chức năng hệ thống cần được mở rộng và hoàn thiện hơn nữa.
   * -  Gặp nhiều bug khi triển khai, mất nhiều thời gian để tim tòi và fix.
2. Hướng phát triển của đề tài :
   * -  Phát triển đề tài với quy mô rộng hơn.
   * -  Tối ưu hóa cơ sở dữ liệu.
   * -  Hoàn thiện và mở rộng thêm các chức năng của hệ thống nhiều hơn nữa.
   * -  Thiết kế giao diện bắt mắt, thân thiện với người dùng hơn nữa.
   * – Thêm realtime comment like, chức năng chat kết mạng
   * – Phát triển như một mạng xã hội.

**TÀI LIỆU THAM KHẢO**

Danh mục các tài liệu tham khảo:

1. Document reactJS <https://reactjs.org/>
2. Document Redux <https://redux.js.org/introduction/getting-started>
3. Node Js + thư viện express
4. Khoá học trên NodeJS + ReactJs trên <https://www.udemy.com/>
5. Một số trang web khác như: topdev, viblo,….